

CA-IDMS[®] Visual DBA

Getting Started
1.0



Computer Associates[™]

This documentation and related computer software program (hereinafter referred to as the "Documentation") is for the end user's informational purposes only and is subject to change or withdrawal by Computer Associates International, Inc. ("CA") at any time.

THIS DOCUMENTATION MAY NOT BE COPIED, TRANSFERRED, REPRODUCED, DISCLOSED OR DUPLICATED, IN WHOLE OR IN PART, WITHOUT THE PRIOR WRITTEN CONSENT OF CA. THIS DOCUMENTATION IS PROPRIETARY INFORMATION OF CA AND PROTECTED BY THE COPYRIGHT LAWS OF THE UNITED STATES AND INTERNATIONAL TREATIES.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR DAMAGE.

THE USE OF ANY PRODUCT REFERENCED IN THIS DOCUMENTATION AND THIS DOCUMENTATION IS GOVERNED BY THE END USER'S APPLICABLE LICENSE AGREEMENT.

The manufacturer of this documentation is Computer Associates International, Inc.

Provided with "Restricted Rights" as set forth in 48 C.F.R. Section 12.212, 48 C.F.R. Sections 52.227-19(c)(1) and (2) or DFARS Section 252.227.7013(c)(1)(ii) or applicable successor provisions.

© 1998 Computer Associates International, Inc., One Computer Associates Plaza, Islandia, New York 11788-7000. All rights reserved.

All product names referenced herein belong to their respective companies.

Contents

Chapter 1: Welcome!

| | |
|--|-----|
| Welcome to CA-IDMS Visual DBA! | 1-1 |
| Features of CA-IDMS Visual DBA | 1-1 |
| CA-IDMS Objects That You Can Manage | 1-2 |
| Manipulating CA-IDMS Objects | 1-4 |
| Viewing and Maintaining Multiple CA-IDMS Systems | 1-4 |
| Managing Objects Using Drag-and-Drop | 1-5 |
| Increasing Productivity | 1-5 |
| Testing SQL Queries | 1-5 |
| Customizing the User Interface | 1-5 |
| What You Need to Know | 1-6 |
| Getting Help | 1-6 |
| Related Documentation | 1-6 |

Chapter 2: Installing and Starting CA-IDMS Visual DBA

| | |
|--|-----|
| Software Requirements | 2-1 |
| Installing CA-IDMS Visual DBA on the PC | 2-1 |
| Installing CA-IDMS Visual DBA on the Mainframe | 2-1 |
| Beginning Your DBA Session | 2-3 |
| Starting CA-IDMS Visual DBA | 2-3 |
| Establishing Communications | 2-4 |
| Opening a Database Object Manager Window | 2-6 |
| Securing CA-IDMS Visual DBA Objects | 2-7 |
| Restricting Access to Objects | 2-7 |
| Assigning Privileges to View Object Instances | 2-8 |
| Assigning Privileges to Update Object Instances | 2-8 |
| Assigning Privileges to Create, Alter, and Drop Object Instances | 2-9 |
| Assigning Privileges to Grant and Revoke Object Privileges | 2-9 |
| What's Next | 2-9 |

Chapter 3: A Tour of the Database Object Manager Window

| | |
|---|------|
| In This Chapter | 3-1 |
| Expanding and Collapsing the Object Tree | 3-2 |
| Expanding the Tree | 3-2 |
| Collapsing the Tree | 3-3 |
| Changing the Tree Structure | 3-3 |
| Restart from Position | 3-3 |
| Tear-Out Window | 3-4 |
| Scratchpad | 3-5 |
| Manipulating Tree Objects | 3-6 |
| Finding Objects | 3-7 |
| Selecting and Copying Objects | 3-7 |
| Viewing, Executing, and Logging Object Syntax | 3-8 |
| Refreshing the Tree | 3-8 |
| Connecting to Multiple CA-IDMS Systems | 3-9 |
| Setting Display Options for Your Session | 3-9 |
| Creating, Opening, and Saving Environments | 3-10 |
| Trying It Out Yourself | 3-10 |
| What's Next | 3-13 |

Chapter 4: A Tour of the SQL Test Window

| | |
|---|-----|
| Overview | 4-1 |
| The SQL Test Window | 4-1 |
| Creating a Query using the SQL Assistant | 4-2 |
| Running and Managing Queries | 4-2 |
| Setting Preferences for the SQL Test Window | 4-3 |
| Putting the SQL Test Window In Action | 4-3 |

Appendix A: Security TABL Resource Tables

| | |
|--|------|
| Overview | A-1 |
| Table 1 — Objectnames in Tree Order, Level and Objectid | A-2 |
| Table 2 — Objectnames in Alphabetic Order, Level and Objectid | A-10 |
| Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames | A-18 |
| Table 4 — Resourcename in Alphabetic Order, Level and Objectname | A-26 |

Welcome!

Welcome to CA-IDMS Visual DBA!

CA-IDMS Visual DBA is a robust, comprehensive tool that allows you to manage CA-IDMS database objects from a single, object-oriented graphical user interface. It also allows the database administrator (DBA) to perform essential, everyday tasks quickly and easily — employing an easy-to-use, intuitive, graphical user interface (GUI) under Microsoft Windows 95 or Windows NT.

With CA-IDMS Visual DBA, you no longer need to be concerned with the details of how to administer your database, such as learning the syntax for multiple online compilers. CA-IDMS Visual DBA simplifies the operations that you perform routinely, allowing you to increase your productivity and decrease the learning curve associated with database administration.

Features of CA-IDMS Visual DBA

CA-IDMS Visual DBA was designed as an intuitive graphical-user interface tool with these features:

- CA-IDMS Visual DBA displays all CA-IDMS objects in a hierarchical tree. You can create, alter, drop, and display objects and assign privileges for each object in the tree.
- You can connect to multiple CA-IDMS systems during one CA-IDMS Visual DBA session.
- You can increase productivity and decrease your learning curve because you no longer need to know the syntax of multiple online compilers.
- You can manipulate CA-IDMS objects offline by working in a saved configuration file.
- You can test SQL queries in the SQL Test window.

The user interface for CA-IDMS Visual DBA uses many Windows 95/Windows NT features that should already be familiar to you. These include:

- Drill-down icons

- Popup menus to manipulate objects
- Tear-out windows to customize your view of the hierarchical object tree display
- The ability to drag-and-drop and cut-and-paste objects

CA-IDMS Objects That You Can Manage

CA-IDMS Visual DBA manages a large number of CA-IDMS objects which you currently manage using the system generation, schema, subschema, DMCL, SQL, and dictionary compilers, as well as the Online Command Facility (OCF) and Batch Command Facility (BCF). The table below summarizes the objects you can create, alter, delete, and display using CA-IDMS Visual DBA:

| Object | Sub Objects | |
|---------------------------|--|---|
| Class | Attribute | |
| Database Table | Subschema mapping DMCL Grantee for Dtable | Dbname Dbgroup in Dtable |
| Dictionary | Class | |
| DMCL | Segment Area Journal File and Area override | Files Buffer Grantee |
| IDD Record | Element Synonym | |
| Non-SQL Schema | Area Set Grantee for Use Registered User Class/Attribute | Record Subschema Comment Responsible User |
| Relational Command Module | | |
| Security | Central user Group Application and category | Dictionary user Profile Grantees for administration |
| Segment | Area DMCL Schema in Segment | File Dbname in Segment Grantees for Segment |
| SQL Schema | Table Table procedure Access Module | View Constraint Grantee |

| Object | Sub Objects | |
|--------|-------------------|-----------------|
| System | Autotask | Destination |
| | Key Table | Line |
| | Loadlist | Map Table |
| | Node | Program |
| | Queue | Resource table |
| | Run Unit | Storage pool |
| | XA Storage pool | Task |
| | Grantee | Lterm |
| | Printer | |
| | | |
| Table | Index | View |
| | Constraint tables | Access module |
| | Calc key | Column |
| | Check condition | Grantee |
| View | View component | Access module |
| | Column | View definition |
| | Grantee | |

Manipulating CA-IDMS Objects

CA-IDMS Visual DBA presents your database objects using an “object-oriented” approach. We perceive the world as a collection of objects that interact with each other. For most database object types, you can select it and right-click to display a popup menu that allows you to create, alter, and drop the object, and assign privileges to it.

Viewing and Maintaining Multiple CA-IDMS Systems

CA-IDMS Visual DBA is designed to make you a more productive and responsive database administrator. One of the ways that you can accomplish this is by connecting to multiple CA-IDMS systems simultaneously and customizing the view of data for each — in its own *Database Object Manager* window.

Another major benefit that CA-IDMS Visual DBA provides is its ability to save an environment that you have configured. All windows that are opened in your workspace — along with the data within them, your connections, and certain environment settings — are saved in a configuration file that can be opened in a future session. This saves you time and trouble while allowing you to immediately begin your activities after starting CA-IDMS Visual DBA.

Managing Objects Using Drag-and-Drop

In addition to presenting the database environment in a graphical format, CA-IDMS Visual DBA provides the ability to copy object definitions from one subbranch to any other that has objects of the same type; this includes copying definitions from one IDMS central version or system to another.

You simply select an object or group of objects in the same subbranch, and then drag and drop it to the target database. CA-IDMS Visual DBA does the appropriate copy and verifies that any integrities involved in the entities are properly maintained.

Using drag-and-drop techniques, you can greatly simplify database administration tasks, especially for test and production database maintenance.

Increasing Productivity

With CA-IDMS Visual DBA, you can manage your CA-IDMS systems simply pointing and clicking. This saves precious time, because there is no longer any need to remember lengthy command syntax and switch settings.

Testing SQL Queries

A large part of the DBA's role is to query a database. CA-IDMS Visual DBA has a tool that lets you create and test your SQL queries. Known as the SQL Test window, this tool includes a query editor and an SQL Assistant, which helps you create queries. When you test a query, CA-IDMS Visual DBA automatically displays the query results in the SQL Test window.

Customizing the User Interface

Like other Windows products, CA-IDMS Visual DBA provides many customization features that allow you to tailor the environment to suit your needs. Not only can you configure such options as the fonts used in various windows, refresh settings, the status bar display, and others, but more importantly, you can control how your data is displayed.

For example, you can define your own custom view of a database. You can re-orient the view to see the system from the perspective of any other object. This feature allows you to easily access the information you need to view and maintain.

What You Need to Know

This guide assumes that you are already familiar with the concepts and features associated with your CA-IDMS installation. If you are unfamiliar with these concepts, please refer to your CA-IDMS documentation for more information.

In addition to an understanding of CA-IDMS concepts and features, this guide assumes that you are familiar with Windows terminology and navigational techniques. This includes how to work with standard Windows items like menus, dialog boxes, the Clipboard, and the Control Panel.

If you are unfamiliar with Windows standards, please refer to your Windows documentation before using CA-IDMS Visual DBA.

Getting Help



CA-IDMS Visual DBA provides online help, which can be used to display information on your console as you work. From the CA-IDMS Visual DBA workspace environment, you can also press the F1 key for context-sensitive help on menu commands, dialogs, and active windows.

Related Documentation

The entire CA-IDMS document set is viewable on the CA-IDMS Documentation CD. The following documents, in addition to this one and the online help, may be the most useful to you when you use this product:

- *IDD DDDL Reference* for information about class and attribute definitions and IDD record definitions
- *Database Administration* for information about defining DMCLs, schemas, and subschemas
- *SQL Reference* for information about defining SQL schemas, tables, and views
- *System Generation* for information about defining the components of a CA-IDMS system
- *Security Administration* for information about assigning privileges to users

Installing and Starting CA-IDMS Visual DBA

Software Requirements

Mainframe Requirements

CA-IDMS Visual DBA runs on any mainframe platform that supports CA-IDMS Server. To run CA-IDMS Visual DBA, you need:

- CA-IDMS Release 12.01 gen level 9506 or later
- CA-IDMS SQL Option
- CA-IDMS Server Release 2.0 or above

PC Requirements

To run CA-IDMS Visual DBA on a PC, you need:

- Microsoft Windows 95 or Windows NT version 3.51 or 4.0
- CA-IDMS Server 4.0 or later versions

Installing CA-IDMS Visual DBA on the PC

To install CA-IDMS Visual DBA on the PC, run **install.exe** from the installation disk. The installation program guides you through the installation process.

Installing CA-IDMS Visual DBA on the Mainframe

To install CA-IDMS Visual DBA on the mainframe, you need to:

Edit, upload, and execute an SQL script that defines schemas, views, and table procedures that CA-IDMS Visual DBA uses. This script is dependent on the CA-IDMS Release. The name of this script is VDBAR12X.SQL for all 12.x releases and VDBAR14X.SQL for all 14.x releases.

Important! *Your specific environment might require some additional installation steps. You must always check the READMEMF.TXT file in the mainframe directory of CA-IDMS Visual DBA on your PC workstation. This document also contains more detailed mainframe installation instructions.*

Edit, Upload, and Execute VDBARxxx.SQL

The VDBARxxx.SQL file contains the SQL definition for table procedures, schemas, and views used by CA-IDMS Visual DBA. It is installed into the CA-IDMS Visual DBA mainframe directory on your PC.

Using a text editor of your choice, open the VDBARxxx.SQL file. In the CREATE TABLE PROCEDURE SYSCA.VDBA_DICTIONARY statement, located in section 0, enter the names of the dictionaries that you want CA-IDMS Visual DBA to display to the user. By default, CA-IDMS Visual DBA displays the SYSTEM, APPLDICT, and SYSDICT dictionaries.

Note: You may defer editing the file until after you upload it to the mainframe.

Below is a sample of the VDBARxxx.SQL code to be edited:

```
--
-- Section 0.Dictionaries seen by CA-IDMS Visual DBA
--

--*****
**          CONFIGURATION IS REQUIRED!          **
**                                          **
** Each parameter in the table procedure      **
** SYSCA.VDBA_DICTIONARY identifies a dictionary **
** that will be seen as a dictionary instance **
**                                          **
-- *****
DROP TABLE PROCEDURE SYSCA.VDBA_DICTIONARY;
CREATE TABLE PROCEDURE SYSCA.VDBA_DICTIONARY (
    "SYSTEM"    CHAR(1)
    , "APPLDICT" CHAR(1)
    , "SYSDICT"  CHAR(1)
) EXTERNAL NAME VDBADICT;
```

Note: If the name of the dictionary that contains the IDMSNTWK, IDMSSECU, and IDMSSECS schemas is *not* SYSDIRL, you must replace occurrences of SYSDIRL in section 1 of the VDBARxxx.SQL file with the name of the dictionary used at your site.

Next, upload the file to the mainframe. Then execute the Batch Command Facility, using VDBARxxx.SQL as input, against all dictionaries that will be visible from CA-IDMS Visual DBA.

Note: You may need to increase the catalog size to accommodate the added views and tables. This may be especially true for the CATSYS segment files if you used the original installation defaults. You should have at least 1000 pages in your DDLCAT area for all catalogs to be used with CA-IDMS Visual DBA.

Beginning Your DBA Session



Once you have started Windows and CA-IDMS, you can begin your work session by starting CA-IDMS Visual DBA and connecting to a CA-IDMS system.

Starting CA-IDMS Visual DBA

Windows 95 and
Windows NT 4.0

In Windows 95 or Windows NT 4.0, you can start CA-IDMS Visual DBA by selecting CA-IDMS Visual DBA through the Windows Start menu.

To start the CA-IDMS Visual DBA application, follow the steps below:

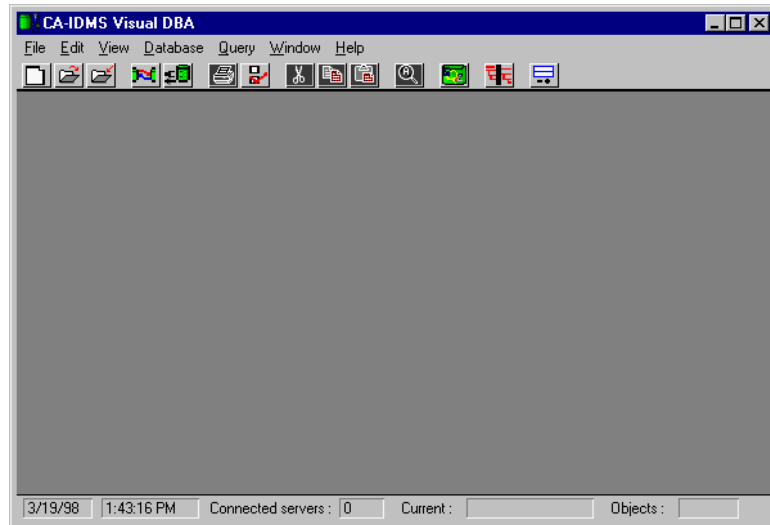
1. From the Start menu, choose Programs.
2. Choose the CA-IDMS Visual DBA submenu.
3. Click CA-IDMS Visual DBA.

Windows NT 3.51

If you are working in Windows NT 3.51, double-click the CA-IDMS Visual DBA icon in your program group, which was created when you installed CA-IDMS Visual DBA.

The Application Window

CA-IDMS Visual DBA is loaded and appears in its own application window. This main window is referred to as the *workspace environment*:



This window initially appears empty; however, once you connect to a CA-IDMS system, the Database Object Manager window appears with Dictionary at the root level. This window is the primary workspace for your tasks.

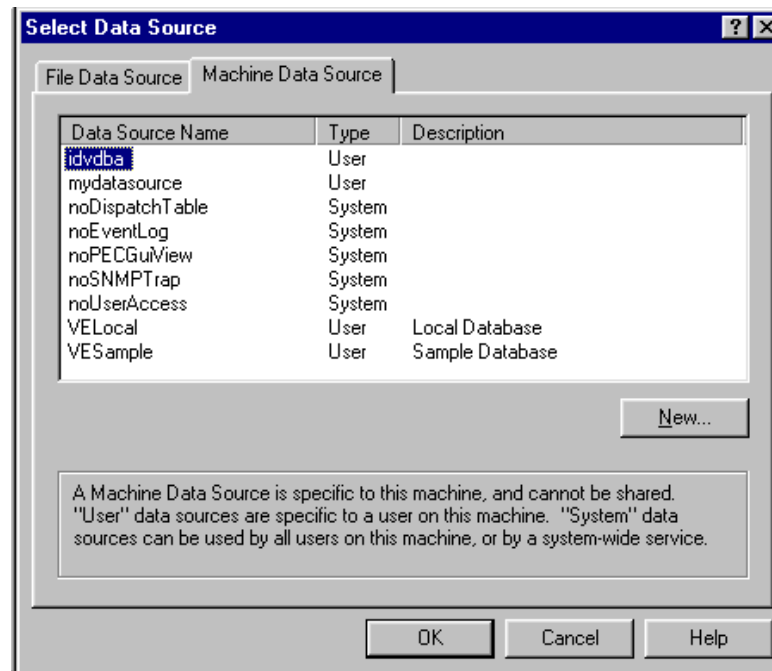
Establishing Communications

After you've started CA-IDMS Visual DBA, connect to a CA-IDMS system to begin your DBA tasks. To accomplish this, you must create or select a CA-IDMS ODBC data source.



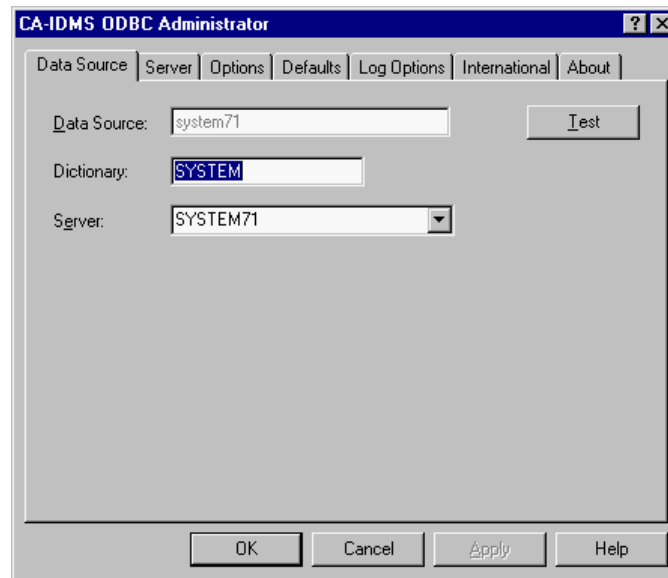
To create or select a CA-IDMS ODBC data source, click Connect from the File menu, or click the Connect toolbar button.

CA-IDMS Visual DBA allows you to choose from a list of existing data sources:



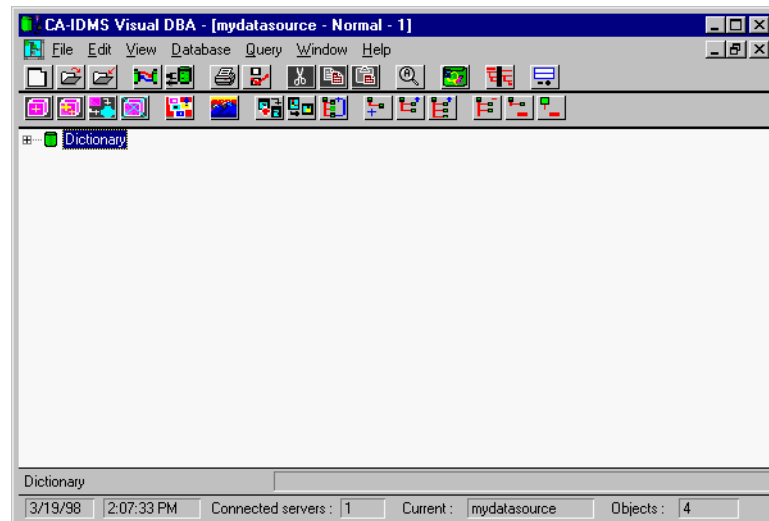
Select a data source and click OK. CA-IDMS Visual DBA issues a logon box for the node. Enter your user ID and password, if required and click OK.

To create a new data source, click New in the Select Data Source dialog. CA-IDMS Visual DBA invokes the CA-IDMS ODBC Administrator where you name the new data source. For the dictionary name, enter the name of the dictionary to which you connected to execute the VDBADDL.SQL file under the Batch Command Facility; that is, the dictionary that contains the table procedure you installed. From the server list box, select a CA-IDMS system. Below is an example:



Opening a Database Object Manager Window

After you have selected a data source, CA-IDMS Visual DBA opens a Database Object Manager window, which will look something like the one below:



After the initial connection, you can use the above procedure to open multiple windows for the same data source or to connect to a different data source.

Note: If you choose to reconnect to the same data source, a new window will be opened.

Securing CA-IDMS Visual DBA Objects

CA-IDMS Visual DBA uses CA-IDMS Server and CA-IDMS SQL to access the information stored in the dictionaries. Because of this, CA-IDMS Visual DBA users need to have the necessary authorization for the CA-IDMS resources that are being used.

This section describes the privileges you need to work with objects in CA-IDMS Visual DBA. These securities involve:

- Viewing object instances in the object tree.
- Updating object instances in the object tree.
- Creating, altering, and dropping object instances in the object tree.
- Granting and revoking user privileges, registering privileges for IDD dictionary entities, and recording user responsibilities for IDD dictionary entities.

CA-IDMS Visual DBA users must be granted privileges for resources of type TABL. The resource names for these TABL resource types are of the form *schema-name.table-name*. Appendix A shows the relationship between the CA-IDMS Visual DBA objects and the TABL resources.

Restricting Access to Objects

To restrict access to certain resource names, and thus to certain CA-IDMS Visual DBA objects, use the information in Appendix A to determine the resource names you need to secure. You can use wildcards in many cases to facilitate the security definition.

For example, to grant access for all objects except the View Definition:

1. Lookup View Definition in Table 3 in Appendix A. The resource name for this object is SYSTEM.SYNTAX.
2. Next, check Table 4 which indicates that by securing resource name SYSTEM.SYNTAX, you automatically also restrict access to at least two additional objects — View Definition and Check Condition. This is because the system table SYSTEM.SYNTAX holds information for View Definition and Check Condition.

So, to limit access to all objects except the View Definition, you need to grant access to all resource names listed in Table 4, except SYSTEM.SYNTAX, which also means that access to Check Condition is also restricted.

Keep in mind that, when limiting access to an object, normally all access to children of the object is also restricted. You do not need to explicitly restrict access to the child resource names. For example, when you restrict access to the SYSVNTWK.SYSTEM resource name, you restrict access to the System object and all of the child objects, such as Program, Queue, Line, .and so on.

Some resource names refer to specific non-SQL schemas as shown in this table:

| Resource Name | Non-SQL Schema |
|---------------|----------------|
| IDMSSECS.* | IDMSSECS |
| IDMSSECU.* | IDMSSECS |
| SYSTSCHM.* | IDMSNTWK |
| SYSVNTWK.* | IDMSNTWK |

If access is granted to one of these resources, you also need to grant Use privilege for the corresponding Non Sql schema. For more information, see the SQL definitions in the CA-IDMS Visual DBA member VDBADDL in your CA-IDMS source library.

Assigning Privileges to View Object Instances

Tree navigation is a pure retrieval operation. To expand the whole tree in the Database Object Manager window, you must minimally have SELECT privilege for all the resource names of type TABL listed in Table 4 in Appendix A.

Assigning Privileges to Update Object Instances

CA-IDMS Visual DBA uses the CA-IDMS compilers (SCHEMA, SUBSCHEMA, IDD, SYSGEN and BCF/OCF) to perform updates to object definitions. It runs these compilers with the help of table procedure SYSCA.VDBAIDD. To perform any updates with CA-IDMS Visual DBA, you need the SELECT privilege on resource name SYSCA.VDBAIDD.

Similarly, to globally prohibit updates of any of the CA-IDMS Visual DBA objects, you can revoke the SELECT privilege on resource name SYSCA.VDBAIDD for a user.

Assigning Privileges to Create, Alter, and Drop Object Instances

To create an object instance by altering an existing one, or to alter or drop an object instance, you must be able to view the object and any related objects that might appear in lists in Create/Alter dialogs. To create an object instance without altering an existing one, you do not need to be able to view the object.

To execute the syntax generated by CA-IDMS Visual DBA for the CA-IDMS compilers, you must be able to pass all standard CA-IDMS security checks. In this way CA-IDMS Visual DBA does not differ from any compiler invoked directly on a CA-IDMS system; that is, to Drop an instance of the DMCL object, you need to have Drop privileges for that DMCL instance.

Assigning Privileges to Grant and Revoke Object Privileges

To invoke the security dialogs, you need to have the authorization to view the grantee (Central User, Dictionary User and Group), and the instances of the granted object.

To execute the security syntax generated by CA-IDMS Visual DBA for the CA-IDMS compilers, you need to pass all standard CA-IDMS security checks. In this way CA-IDMS Visual DBA does not differ from defining security through any direct invocation of BCF/OCF or IDD.

What's Next

Now that you have learned how to start CA-IDMS Visual DBA, connect to a server, open the Database Object Manager window, and secure its objects, you can perform a variety of database administration tasks. The next chapter introduces you to the features of the Database Object Manager Window.

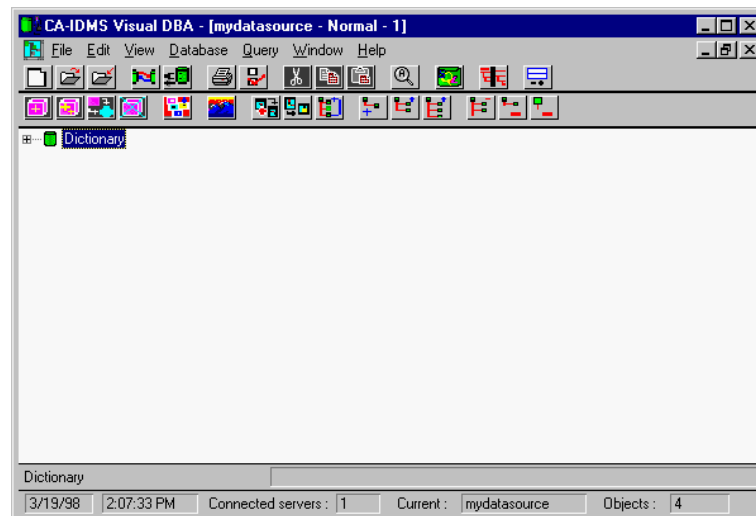
A Tour of the Database Object Manager Window

In This Chapter

At the heart of CA-IDMS Visual DBA, the Database Object Manager provides a convenient and organized way to view and manipulate the database information that is currently stored on a particular server.



The Database Object Manager window appears after you connect to a CA-IDMS database:



Initially, the Database Object Manager displays the *root* level Dictionary branch. From the root level, you can drill down to the information you need.

Using the Database Object Manager, you can create, alter, drop, and display CA-IDMS database objects, and assign privileges to them. Additionally, you can open and simultaneously work with multiple databases and/or servers in multiple windows.

Expanding and Collapsing the Object Tree

The information in the Database Object Manager window is arranged in a tree structure that clearly shows the relationships among the pieces of information on a server. This Database Object Manager tree initially displays a group of categories or *branches*. The tree is collapsible and expandable—to view related information, merely double-click on the branch of your choice.

Using this tree, you can manipulate information. For example, you may alter an object's characteristics or change the properties associated with an object.

As you start to add, modify, or delete objects on the server, the Database Object Manager will display the latest information.

Note: You can have more than one Database Object Manager window open at the same time, with each window connected to the same or different data sources:

Expanding the Tree



Since the ability to expand and collapse the tree branches in the Database Object Manager window is an important feature of CA-IDMS Visual DBA, it is a good idea to familiarize yourself with the different ways this can be accomplished.

One Level at a Time



To expand a branch a single level, click on its expansion button (⊞), which displays the sub-branches that exist one level down within the selected branch. (Note that only expandable branches have this button.) You may also double-click anywhere on the branch to expand it. In addition, it is possible to expand a branch by using toolbar buttons, the Plus (+) key, or the menu commands on the View menu.

Expanding Multiple Levels



You can also use the toolbar, control keys, and the View menu when you want to see all the sub-branches under a single branch. Simply select the branch and click on the Expand Branch toolbar button, press the Multiply (*) key, or use the Expand Branch command from the View menu.

Similarly, to view all sub-branches for *every* branch, click on the Expand All Branches toolbar button, press Ctrl + *, or choose the Expand All Branches command from the View menu.

Infinite Drill

One of the most useful aspects of CA-IDMS Visual DBA is its “infinite drill” feature which graphically illustrates the complex relationships that exist among objects. By expanding – or drilling down through – branches in the tree, you will see that nested within most new sub-branches are lower-level sub-branches that contain related information.

Further, combining infinite drill-down with the features described in the Changing the Tree Structure section, allows you to turn *any* sub-branch into a root branch from which you can drill down.

Collapsing the Tree



To collapse a sub-branch, choose the collapse button (≡) to the left of the object category name or use the appropriate toolbar buttons or menu commands from the View menu.

Changing the Tree Structure

To facilitate your ability to shift your perspective of the database at the touch of a button, CA-IDMS Visual DBA offers three distinct features: Restart from Position, Tear Out and Scratchpad.

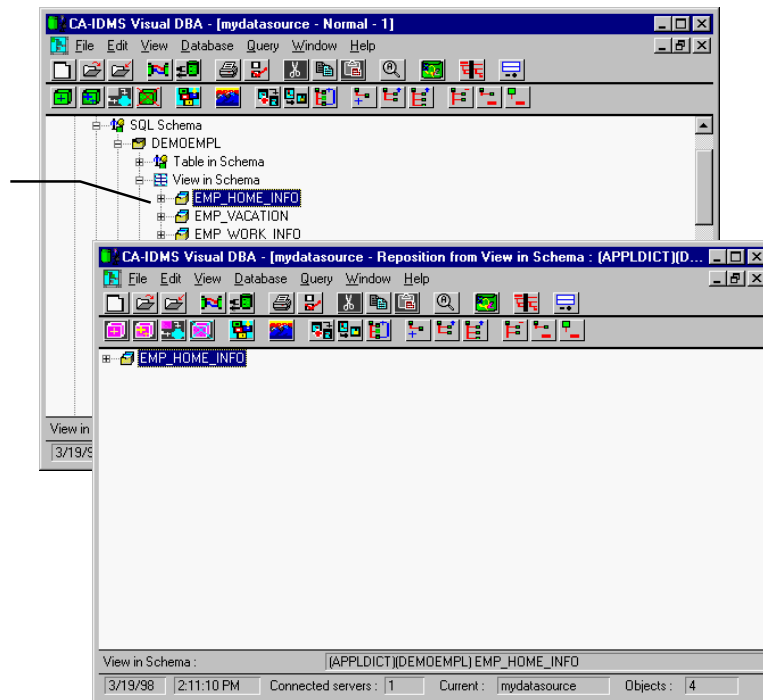
Restart from Position



The information you need often begins on a level far beneath the root object category branch. The layers of sub-branches that you must pass through may be extraneous, in this instance, so you may want to remove them from view. To do this, use the Restart From Position toolbar button to execute the command that converts any branch you select into the root branch of your window.

For example, in the following window, the branch, “EMP_HOME_INFO” is nested deep within the tree. Clicking on the Restart from Position toolbar button changes the window, as follows:

Click Restart from Position on the Windows menu to display deeply nested objects at the root level.



Now, “EMP_HOME_INFO” is the root branch and may be expanded using the same techniques previously described.

Tear-Out Window

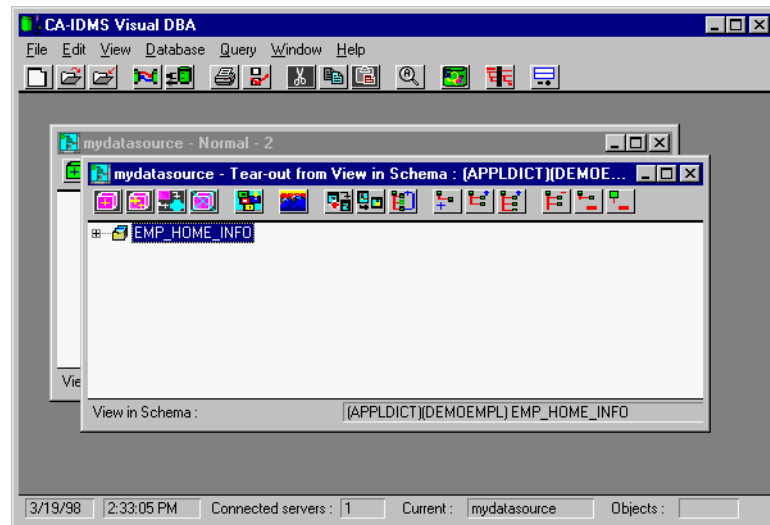


You have seen how the Restart From Position command on the Windows menu alters the existing Database Object Manager window. If you want to view a branch in a *new* window, you may use the Tear-out feature.

By selecting a branch and clicking the Tear Out toolbar button (or choosing the Tear Out command from the Window menu), you can create a new window, while leaving the original window intact. The branch you select becomes the root branch in the new window and may be expanded to reveal all existing sub-branches.



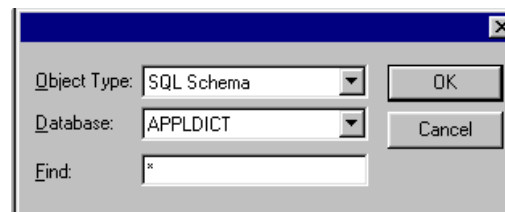
As an example, let’s look at the Database Object Manager window that was used in the previous section. By clicking on the Tear Out toolbar button, a new Database Object Manager window is opened, with “EMP_HOME_INFO” as the root branch:



Scratchpad

The Scratchpad command on the Window menu offers yet another powerful way to construct your own customized Database Object Manager window. The Scratchpad window is empty when it is first opened.

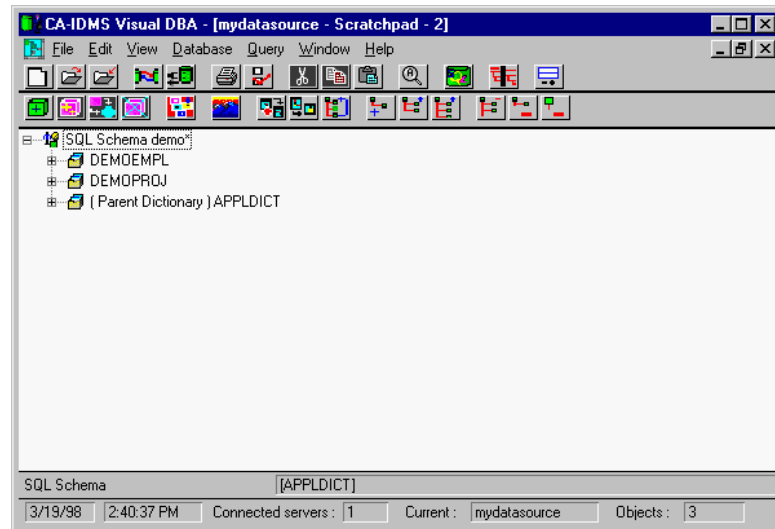
CA-IDMS Visual DBA allows you to easily find any object on the current server and display it as the root branch in the Scratchpad window. To do this, open a Scratchpad window and then choose the Edit Locate menu command, which invokes the Locate Object dialog box:



In the Object Type drop-down list box, choose the object category that you wish to find. In the Find edit control, leave the asterisk (*) unchanged if you want to locate the object category branch. Otherwise, enter the full name of an object (or a wildcard expression).

For example, you may want to find the SQL schemas that start with "demo". To do this, select SQL Schema from the Object Type drop-down list box. Next, enter **demo*** in the Find edit control, and choose OK. A single branch labeled "SQL Schema demo*." appears in the Scratchpad window.

Expanding this branch reveals the branches for SQL schemas whose names start with DEMO:



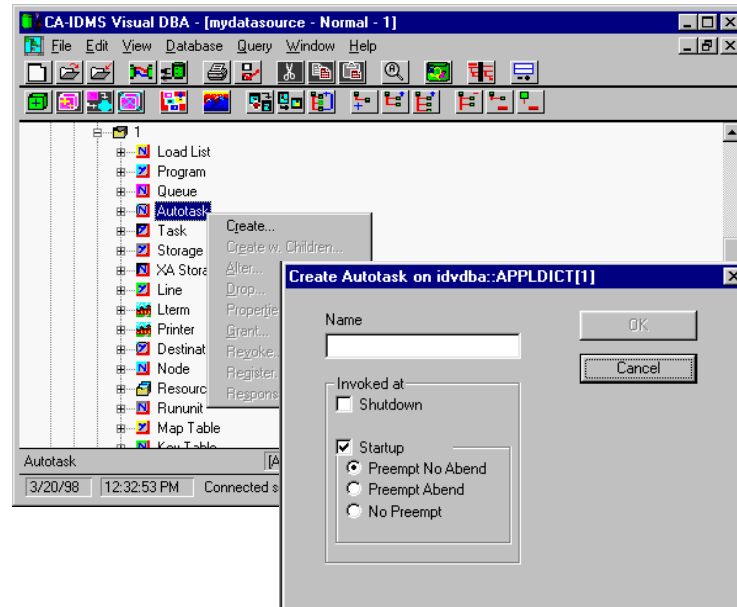
The Window Restart From Position, Tear Out, and Scratchpad menu commands all enable you to construct your own Database Object Manager windows that focus precisely on the information *you* need.

Manipulating Tree Objects

Once you navigate your way to the object in the tree that you want to manipulate, you can create a new object entity, modify an existing one, delete it, display it, and define privileges for it.

To do these things, select the object and right-click. A popup menu appears which displays the actions allowed for the object. Alternatively, you can select the object and use the commands on the Edit menu.

In this screen shot, the Autotask system object is selected. The popup-menu offers the option of creating a new system autotask. When you click Create, a dialog appears that guides you through the Autotask definition. If you need help understanding the dialog options, press F1 for detailed field-level help.



Finding Objects



CA-IDMS Visual DBA offers a search tool to search for an object or object category in an expanded tree. For example, if you want to find a specific table, click the Find button (or use the Edit Find command). CA-IDMS Visual DBA displays a dialog that prompts you for the search string and other search criteria.

Selecting and Copying Objects

To select multiple objects in the same branch of a tree:

- Click the left-mouse button and hold the Ctrl key to select a non-contiguous group of objects
- Click the left-mouse button and hold the Shift key to select a range of objects

You can also copy object definitions from one subbranch to any other that has objects of the same type; this includes copying definitions from one IDMS central version or system to another.

To copy the object definitions, simply select the objects, and drag and drop them on to the target database. CA-IDMS Visual DBA does the appropriate copy and verifies that any integrities involved in the entities are properly maintained.

Drag-and-drop techniques simplify database administration tasks, especially for test and production database maintenance.

Viewing, Executing, and Logging Object Syntax

CA-IDMS Visual DBA lets you decide what happens when you create, alter, or delete an object and then click OK. You can view, execute, and log the syntax that CA-IDMS Visual DBA creates.

To choose the action that occurs when you click OK, select Preferences from the File menu and then click the OK Actions icon in the Preferences window. For each CA-IDMS compiler (IDD, OCF/BCF, schema, subschema, or sysgen), select one or more check boxes to log, view, and execute syntax.

By default, CA-IDMS Visual DBA executes the object syntax and displays the syntax only if an error occurs. If you select View Syntax only, it displays, but does not execute the syntax. If you select View Syntax and Exec Syntax, it displays the syntax and asks if you want to execute it.

If you select Log Syntax, CA-IDMS Visual DBA creates a log file with an extension that designates the compiler and logs all object syntax for that compiler to the log file. Later, you can open the log file, edit it if you want to, and then upload and execute it as a batch file on the mainframe.

Refreshing the Tree



With the Force Refresh command on the View menu or toolbar, you can refresh data "on-the-fly" for either a single branch, all branches, or any object type.

The alternative to the Force Refresh command is Activate Background Refresh, also on the View menu. This command instructs CA-IDMS Visual DBA to refresh the data at a specified frequency, using the parameters defined in the Refresh dialog on the File Preferences command.



Typically, CA-IDMS Visual DBA refreshes the object tree in the background at a specified frequency. You can use the Refresh option in the File Preferences command to set different refresh times for individual objects.

You can also save your refresh setting as default settings to use with current and new workspace environment configurations.

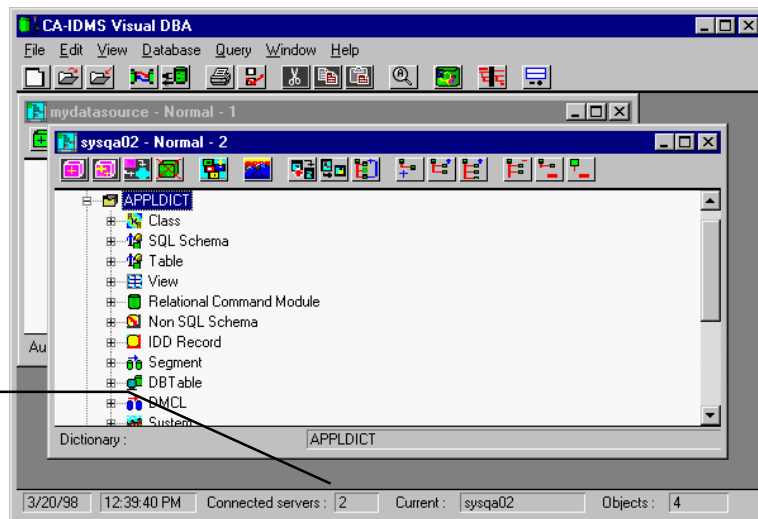
Connecting to Multiple CA-IDMS Systems



To connect to more than one CA-IDMS system, simply click the Connect button to choose another CA-IDMS server to connect to.

CA-IDMS Visual DBA opens another Database Object Manager window which displays the object tree data for the CA-IDMS system that you just connected to. Using the options on the Windows menu, you can display tile and cascade the windows to view them:

The statusbar shows how many servers you have connected to.



When you connect to multiple CA-IDMS Visual DBA sessions, you can use the Sessions option in the File Preferences command to limit the number of sessions you can run at one time

Setting Display Options for Your Session

To customize your CA-IDMS Visual DBA environment even more, you can use options found under File Preferences to select displays options for your sessions. These include:

- Choosing the font and colors of the toolbar balloon help.
- Selecting what you want to display in the status bar
- Selecting font styles to display in the Database Object Manager window, SQL Test window, and status bar
- Selecting whether to display the status bar and toolbar

Creating, Opening, and Saving Environments



Because CA-IDMS Visual DBA gives you so many ways to customize your session options and object tables, naturally, it lets you save your customized environments so that you do not have to re-define them each time you start a session.

For example, suppose you ‘tear-out’ a branch of the object tree and display it in a new window. You can save that view of the tree in a configuration (.CFG) file by using the File Save As command.

The next time you want to use that view of the object tree, you can open the configuration file using the File Open command. Similarly, you can create a new configuration file by clicking File New.

Trying It Out Yourself

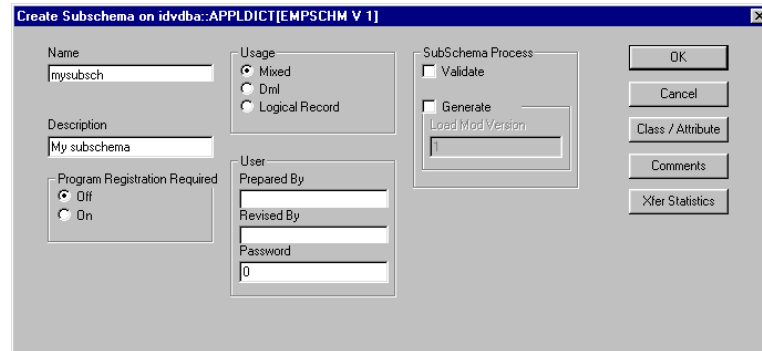
Now let’s apply the features of the Database Object Manager to work. In this example, we will create and modify a non-SQL subschema based on an existing subschema definition. To do this, we will use:

- A tear-out window
- The drag-and-drop feature to copy an existing subschema area to the new subschema definition
- The Create and Alter object options

First, open CA-IDMS Visual DBA and connect to a CA-IDMS data source that contains the Commonwealth demo database supplied with CA-IDMS installation. After logging on, drill down the root structure to the subschema object by double-clicking the:

- Dictionary object
- The APPLDICT dictionary object
- Non-SQL schema object

Next, right click the Subschema object and select Create. In the Create Subschema dialog, enter a name for your subschema, such as MYSUBSCH and then click OK:



CA-IDMS Visual DBA refreshes the Database Object Manager and displays the name of the subschema you just created.

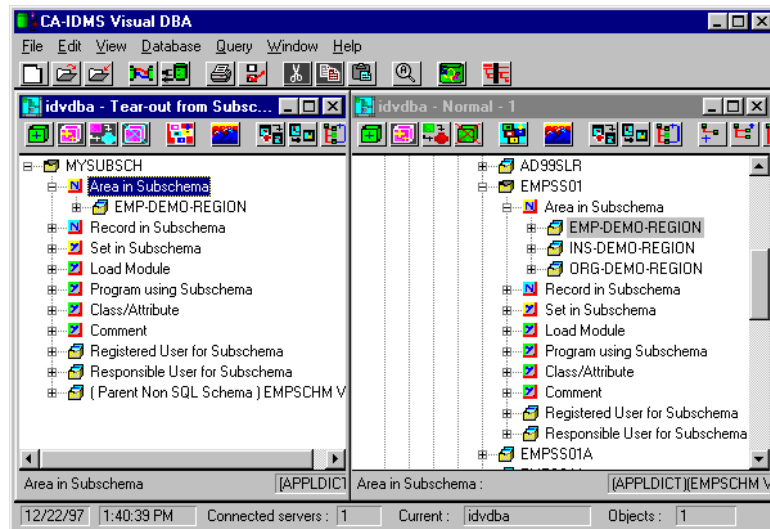


Now, let's populate the new subschema with information copied from an existing subschema. To make it easier to view, first 'tear-out' the subschema you created and display it in a new window. To do this, select the subschema, MYSUBSCH, and click Tear Out from the Window menu. A new window appears with your subschema at the root level. Next, click Tile Vertical button to display both windows side-by-side.

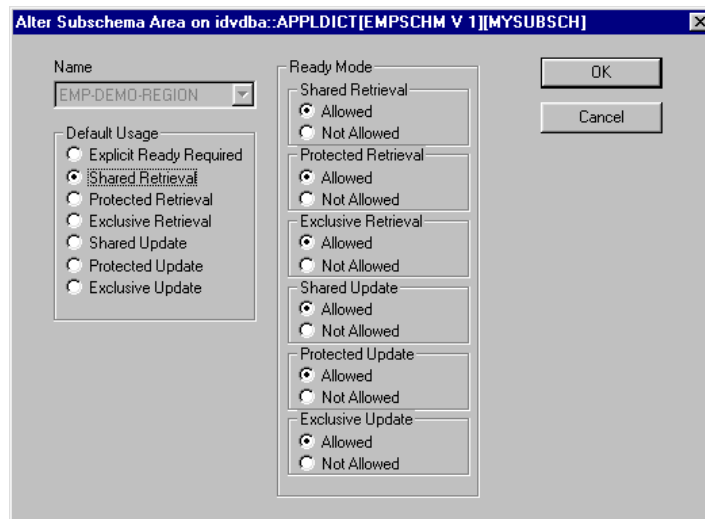
In the original window, double-click the EMPSS01 subschema and then double-click Area in Subschema. We are going to copy the EMP-DEMO-REGION area to the subschema you just created. In the 'Tear-Out' window, double-click MYSUBSCH to display the subschema objects.

To copy the EMP-DEMO-REGION area to the new subschema, drag and drop EMP-DEMO-REGION from the original window to the Area in Subschema object of the 'Tear-Out' window. Notice that the mouse cursor changes to the Area icon when it is positioned over Areas in Subschema.

If you successfully copied the area, your window should look something like this:



Finally, let's modify the default usage of the EMP-DEMO-REGION area in the new subschema. To do this, select it and right-click. From the popup menu, click **Alter**. Click **Shared Retrieval** under **Default Usage** and then click **OK**:



You have just completed a brief tour of the Database Object Manager. You have seen how easily you can use simply point-and-click techniques to create, copy, and modify database objects.

What's Next

This chapter provided you with an overview of the Database Object Manager window and how to use it to display CA-IDMS objects and manipulate them. Now that you have a general understanding of the Database Object Manager window, let's move on to the SQL Test window.

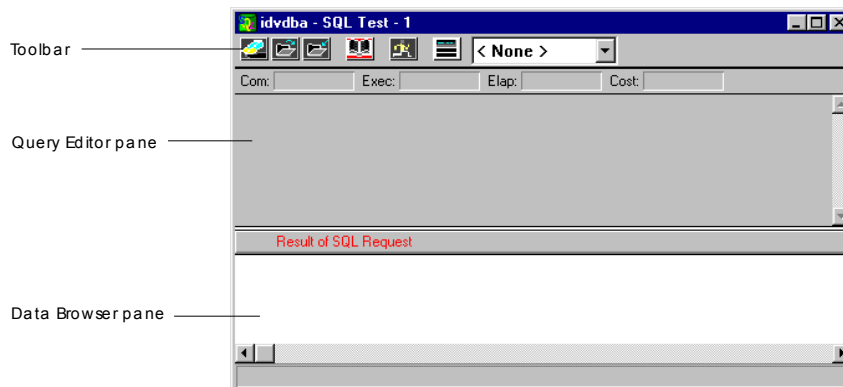
A Tour of the SQL Test Window

Overview

CA-IDMS Visual DBA has a tool that lets you create, edit and test your SQL queries. Known as the SQL Test window, this tool makes it easy to create complex queries against your database by providing an SQL Assistant and familiar Windows editing tools. To open an SQL Test window, click SQL Test from the Database menu.

The SQL Test Window

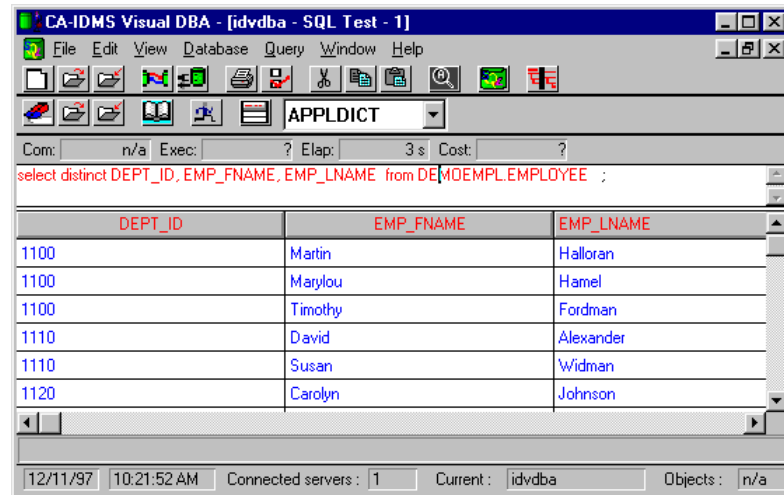
The SQL Test window is arranged in a highly-intuitive manner. A sample window is displayed below:



A handy toolbar at the top of the window contains buttons for clearing the query editor pane, as well as for opening, saving, and executing queries, and updating the query result. An additional button invokes the SQL Assistant, a CA-IDMS Visual DBA tool that helps you build an SQL query if you would prefer not to enter one manually. The database list box allows you to choose the database to which you want to connect.

Because the SQL Test window is divided into two smaller windows tiled horizontally, it is easy to see both your query (in the SQL Query Editor), and the result of your query (in the Data Browser). The two panes in the SQL Test window may be sized according to your own preferences. Also, using the mouse, you may rearrange and resize the columns of data resulting from a query.

After running a Select statement query, the SQL Test window will look similar to the sample that follows:



Creating a Query using the SQL Assistant

To create an SQL query using the SQL Assistant, first open the SQL Test window by selecting SQL Test from the Database menu. CA-IDMS Visual DBA prompts you to select a CA-IDMS database server to connect to.



After connecting, select a database from the drop-down list box in the toolbar. Then invoke the SQL Assistant by clicking the SQL Assistant button on the toolbar. The SQL Assistant is a wizard-type tool that walks you through the process of creating an SQL query. For example, the first dialog prompts you to select the type of query you want to create — Select, Create, Insert, Delete. Then it displays the tables and views you can choose to query, and so on. When you finish, it displays the completed query in the Query Edit Pane.

Running and Managing Queries



Once you have a query in the Query Edit pane, you can run it, by clicking the Run button on the toolbar. CA-IDMS returns the query results and displays them in the Data Browser pane.

You can edit and save the query displayed in the Query Edit pane. CA-IDMS Visual DBA provides the standard Windows editing (Copy, Cut, and Paste) commands accessible on the toolbar and on the Edit menu. Use the Save As and Open commands on the Query menu to save and open SQL queries. By saving the query, you can open and run it again and again without the need to re-create it.

Setting Preferences for the SQL Test Window



CA-IDMS Visual DBA gives you the options of setting preferences for your SQL testing. Click the Preferences toolbar button and then SQL Test to set testing preferences.

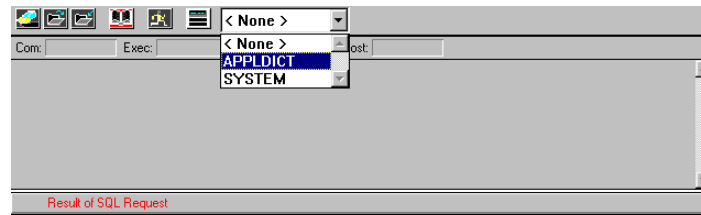
In the SQL Test Preferences dialog, you can select whether to display query results in the Data Browser pane, and if so, how many records CA-IDMS should display. You can also automatically commit any changes made to the data during the execution of the SQL query as well as choose the display font for the SQL Test window.

Putting the SQL Test Window In Action

Now that you are familiar with the features and layout of the SQL Test window, we'll create a query using the SQL Assistant, run it, and save the query we create. We will create a query that displays department names and manager ids, and the names of the employees that work in each department. To do this, we will use the SQL Assistant to select the appropriate tables and columns from the DEMOEMPL schema.

First, start CA-IDMS Visual DBA and click SQL Test from the Database menu. CA-IDMS Visual DBA responds by displaying the Select Data Source dialog where you can connect to an existing CA-IDMS data source or create a new one. Select the datasource that contains the sample Commonwealth SQL database and log on.

In the SQL Test window that appears, select the database to connect to in the drop-down list box on the toolbar. Select dictionary APPLDICT:



Next, click the SQL Assistant button on the toolbar and follow these steps:

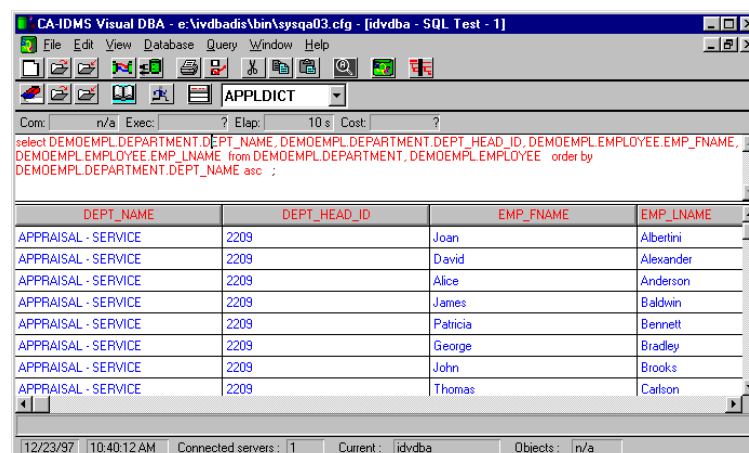
1. Choose Select for the type of SQL statement you want to create and then click Next.
2. For the tables to query, select DEMOEMPL.DEPARTMENT and DEMOEMPL.EMPLOYEE and then click Next.
3. To select the columns for your query, use the drop-down list box to display the columns for each table you selected. Select the columns in the order you want them to appear in the query result.

For our example, click DEPT_NAME and DEPT_HEAD_ID from the DEPARTMENT table. From the drop-down list box, select the EMPLOYEE table and then select EMP_FNAME and EMP_LNAME. Then click Next.

4. CA-IDMS Visual DBA displays a dialog where you can tailor your query by grouping the query results, by specifying search criteria, and so on. In this case, we will simply sort the query results by department name. To do this, click DEMOEMPL.DEPARTMENT.DEPT_NAME in the Order By box and then click Finish.



The query appears in the Query Editor pane. Click the Run button to run the query. After a pause, the query results appear in the Data Browser pane:



From here you can use the edit functions to edit the query that appears in the SQL test window. Once you are satisfied with your results, you can save your query and re-open again and again.

Security TABL Resource Tables

Overview

The tables in this appendix contain the information you need to secure the database objects that you view through CA-IDMS Visual DBA. For more information about securing objects, see Chapter 2, “Installing and Starting CA-IDMS Visual DBA.”

Tables 1 through 4 show the relations between the CA-IDMS Visual DBA objects and the TABL resources. Each table contains:

- The objectname, which is the label of the object as used in the tree. The object names are not unique.
- The objectid, which uniquely identifies an object. For example, objectids GP_GR_SCHEMA and GP_US_SCHEMA both have the same objectname, Schema granted.GP_GR_SCHEMA is the schema granted for a Group, while GP_US_SCHEMA is the schema granted for a User.

The remainder of this appendix displays the contents of tables 1 through 4. These tables contain:

| Table | Contents |
|---------|---|
| Table 1 | Objectname, the level in the tree, and the objectid. The objectnames are listed in the order of the tree, starting with level 0 to level 5. |
| Table 2 | Objectnames ordered alphabetically, level and objectid. |
| Table 3 | Objectname, the level in the tree, and the resourcenames (Schema.table) used by the object. The objectnames are in alphabetic order. |
| Table 4 | Resourcenames (Schema.table) used by the object, the level, and the objectname. The resourcenames are in alphabetic order. |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|---------------------------|-------|---------------|
| Dictionary | 0 | DATABASE |
| SQL Schema | 1 | SCHEMA |
| Table | 1 | ALL_TABLE |
| View | 1 | ALL_VIEW |
| Table Procedure | 1 | ALL_PROCEDURE |
| Relational Command Module | 1 | RCM |
| Non SQL Schema | 1 | NSQLSCHEMA |
| IDD Class | 1 | CLASS |
| IDD Record | 1 | IDDRECORD |
| Segment | 1 | SEGMENT |
| DBTable | 1 | DBTABLE |
| DMCL | 1 | DMCL |
| System | 1 | SYSTEM |
| Category | 1 | CATEGORY |
| Activity | 1 | ACTIVITY |
| System Id | 1 | SYSTEMID |
| System Profile | 1 | SYSPROFILE |
| User Profile | 1 | PROFILE |
| Group | 1 | GROUP |
| Central User | 1 | USER |
| Dictionary User | 1 | DICTUSER |
| Grantee for DCADMIN | 1 | GU_DCADMIN |
| Grantee for SYSADMIN | 1 | GU_SYSADMIN |
| Table in Schema | 2 | TABLE |
| View in Schema | 2 | VIEW |
| Table Procedure in Schema | 2 | PROCEDURE |
| Constraint | 2 | CONSTRAINT |
| Access Module | 2 | AM |
| Grantee for Schema | 2 | GU_SCHEMA |
| Area in Non SQL Schema | 2 | NSQLAREA |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|----------------------------|-------|------------------------|
| Record | 2 | RECORDTYP |
| Set | 2 | NSQLSET |
| Subschema | 2 | SUBSCHEMA |
| Class / Attribute | 2 | NSQLATTRIB |
| Comment | 2 | NSQLCOMMENT |
| Grantee for Use | 2 | GU_NSQLSCHEMA |
| Registree for Schema | 2 | RU_NSQLSCHEMA |
| Responsible for Schema | 2 | SU_NSQLSCHEMA |
| Attribute | 2 | ATTRIBUTE |
| IDD Record Synonym | 2 | IDDRECORDSYN |
| IDD Record Element | 2 | IDDRECORDELEM |
| Area | 2 | AREA |
| File | 2 | FILE |
| DMCL including Segment | 2 | SEGMENT_DMCL |
| DBName in Segment | 2 | SEGMENT_DBNAME |
| Schema referencing Segment | 2 | SEGMENT_SCHEMA |
| Grantee for Segment | 2 | GU_SEGMENT |
| Subschema Mapping | 2 | SSCMAP |
| DBName | 2 | DBNAME |
| DMCL in DBTable | 2 | DBTABLE_DMCL |
| DBGGroup in DBTable | 2 | DBGROUP |
| Grantee for DBTable | 2 | GU_DBTABLE |
| Segment in DMCL | 2 | DMCL_SEGMENT |
| Segment.File Override | 2 | DMCL_SEGMENT_FILE_OVER |
| Segment.Area Override | 2 | DMCL_SEGMENT_AREA_OVER |
| File in DMCL | 2 | DMCL_FILE |
| Area in DMCL | 2 | DMCL_AREA |
| Database Buffer | 2 | BUFFER_DB |
| Journal Buffer | 2 | BUFFER_JOURNAL |
| Archive Journal | 2 | JOURNAL_ARCH |
| Disk Journal | 2 | JOURNAL_DISK |
| Tape Journal | 2 | JOURNAL_TAPE |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|----------------------------------|-------|-----------------------|
| Grantee for DMCL | 2 | GU_DMCL |
| Load List | 2 | LOADLIST |
| Program | 2 | PROGRAM |
| Queue | 2 | QUEUE |
| Autotask | 2 | AUTOTASK |
| Task | 2 | TASK |
| Storage Pool | 2 | STORPOOL |
| XA Storage Pool | 2 | XASTORPOOL |
| Line | 2 | LINE |
| Lterm | 2 | SYSTEM_LTERM |
| Printer | 2 | SYSTEM_PRINTER |
| Destination | 2 | DESTINATION |
| Node | 2 | NODE |
| Local Dictionary in Res Table | 2 | RESTABLE_DBNAME_LOCAL |
| Dictionary via Node in Res Table | 2 | RESTABLE_DBNAME_VIA |
| Destination Node in Res Table | 2 | RESTABLE_DESTNODE |
| Rununit | 2 | RUNUNIT |
| Map Table | 2 | MAPTYPE |
| Key Table | 2 | KEYTABLE |
| Access Module in Category | 2 | CATEGORY_AM |
| Load Module in Category | 2 | CATEGORY_LOADMODULE |
| Program in Category | 2 | CATEGORY_PROGRAM |
| Queue in Category | 2 | CATEGORY_QUEUE |
| Rununit in Category | 2 | CATEGORY_RUNUNIT |
| Task in Category | 2 | CATEGORY_TASK |
| Grantee for Execute | 2 | GU_CATEGORY |
| Grantee for Execute | 2 | GU_ACTIVITY |
| Grantee for System Id | 2 | GU_SYSTEMID |
| Attribute | 2 | SYSATTRIB |
| Grantee on System Profile | 2 | GU_SYSPROFILE |
| Attribute | 2 | ATTRIB |
| Grantee on User Profile | 2 | GU_PROFILE |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|--------------------------|-------|------------------|
| User in Group | 2 | GROUP_USER |
| Access Module granted | 2 | GP_GR_AM |
| Activity granted | 2 | GP_GR_ACTIVITY |
| Area granted | 2 | GP_GR_AREA |
| Category granted | 2 | GP_GR_CATEGORY |
| DBADMIN granted | 2 | GP_GR_DBADMIN |
| DBName & Segment granted | 2 | GP_GR_DBNAME |
| DBTable granted | 2 | GP_GR_DBTABLE |
| DCADMIN granted | 2 | GP_GR_DCADMIN |
| DMCL granted | 2 | GP_GR_DMCL |
| Group granted | 2 | GP_GR_GROUP |
| Non SQL Schema granted | 2 | GP_GR_NSQLSchema |
| Schema granted | 2 | GP_GR_SCHEMA |
| SYSADMIN granted | 2 | GP_GR_SYSADMIN |
| System Id granted | 2 | GP_GR_SYSTEMID |
| System Profile granted | 2 | GP_GR_SYSPROFILE |
| Table granted | 2 | GP_GR_TABLE |
| Table Procedure granted | 2 | GP_GR_PROCEDURE |
| User granted | 2 | GP_GR_USER |
| User Profile granted | 2 | GP_GR_PROFILE |
| View granted | 2 | GP_GR_VIEW |
| Grantee on Group | 2 | GU_GROUP |
| Group User belongs | 2 | USER_GROUP |
| Access Module granted | 2 | GP_US_AM |
| Activity granted | 2 | GP_US_ACTIVITY |
| Area granted | 2 | GP_US_AREA |
| Category granted | 2 | GP_US_CATEGORY |
| DBADMIN granted | 2 | GP_US_DBADMIN |
| DBName & Segment granted | 2 | GP_US_DBNAME |
| DBTable granted | 2 | GP_US_DBTABLE |
| DCADMIN granted | 2 | GP_US_DCADMIN |
| DMCL granted | 2 | GP_US_DMCL |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|-------------------------------|-------|--------------------|
| Group granted | 2 | GP_US_GROUP |
| Non SQL Schema granted | 2 | GP_US_NSQLSHEMA |
| Schema granted | 2 | GP_US_SCHEMA |
| SYSADMIN granted | 2 | GP_US_SYSADMIN |
| System Id granted | 2 | GP_US_SYSTEMID |
| System Profile granted | 2 | GP_US_SYSPROFILE |
| Table granted | 2 | GP_US_TABLE |
| Table Procedure granted | 2 | GP_US_PROCEDURE |
| User granted | 2 | GP_US_USER |
| User Profile granted | 2 | GP_US_PROFILE |
| View granted | 2 | GP_US_VIEW |
| Grantee on User | 2 | GU_USER |
| Entity Type Authority | 2 | GP_DU |
| Non SQL Schema registered | 2 | RP_DU_NSQLSHEMA |
| Subschema registered | 2 | RP_DU_SUBSCHEMA |
| Non SQL Schema responsibility | 2 | SP_DU_NSQLSHEMA |
| Subschema responsibility | 2 | SP_DU_SUBSCHEMA |
| Index | 3 | INDEX |
| View on Table | 3 | TABLE_VIEW |
| Constraint Table referenced | 3 | TABLE_CONST_REFCED |
| Constraint Table referencing | 3 | TABLE_CONST_REFING |
| Access Module for Table | 3 | TABLE_AM |
| Calc Key | 3 | CALC |
| Column | 3 | COLUMN |
| Check Condition | 3 | CHECKCONDITION |
| Grantee on Table | 3 | GU_TABLE |
| View Component | 3 | VIEW_COMPONENT |
| Access Module for View | 3 | VIEW_AM |
| Column of View | 3 | VIEWCOLUMN |
| View Definition | 3 | VIEWDEFINITION |
| Grantee on View | 3 | GU_VIEW |
| Key in Table Procedure | 3 | PROCKEY |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|-----------------------------------|-------|----------------------|
| Parameter of Table Procedure | 3 | PROCCOLUMN |
| Access Module for Table Procedure | 3 | PROCEDURE_AM |
| Grantee on Table Procedure | 3 | GU_PROCEDURE |
| Referencing Table in Constraint | 3 | CONST_TABLE_REFING |
| Referenced Table in Constraint | 3 | CONST_TABLE_REFCED |
| Table accessed | 3 | AM_TABLE |
| View accessed | 3 | AM_VIEW |
| Table Procedure accessed | 3 | AM_PROCEDURE |
| Network Record accessed | 3 | AM_RECORDTYP |
| Grantee on Access Module | 3 | GU_AM |
| Area Procedure | 3 | AREAPROC |
| Record in Area | 3 | AREA_RECORDTYP |
| Record Procedure | 3 | RECORDPROC |
| Data Compression Table | 3 | DATACOMPTAB |
| Record Synonym | 3 | RECORDSYN |
| Record Element | 3 | RECORDELEM |
| Set owned by Record | 3 | RECORDTYP_SET_OWNING |
| Set Record is member | 3 | RECORDTYP_SET_MEMBER |
| Structure Shared By | 3 | RECORDSHARE |
| Owner Record of Set | 3 | SET_RECORD_OWNER |
| Member Record of Set | 3 | SET_RECORD_MEMBER |
| Area in Subschema | 3 | SUBSCHEMA_AREA |
| Record in Subschema | 3 | SUBRECORD |
| Set in Subschema | 3 | SUBSCHEMA_SET |
| Load Module | 3 | LOADMODULE |
| Program using Subschema | 3 | SUBSCHEMA_PROGRAM |
| Class / Attribute | 3 | SUBSCHEMAATTRIB |
| Comment | 3 | SUBSCHEMACOMMENT |
| Registree for Subschema | 3 | RU_SUBSCHEMA |
| Responsible for Subschema | 3 | SU_SUBSCHEMA |
| IDD Element Synonym | 3 | IDDELEMSYN |
| IDD Element Description | 3 | IDDELEMSDES |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|------------------------------|-------|----------------------|
| IDD Element Value | 3 | IDDELEMVALUE |
| IDD Element Indexed By | 3 | IDDNAMEDES |
| File in Area | 3 | AREA_FILE |
| DMCL using Area | 3 | AREA_DMCL |
| Schema using Area as default | 3 | AREA_SCHEMA |
| Table stored in Area | 3 | AREA_TABLE |
| Index stored in Area | 3 | AREA_INDEX |
| Subarea | 3 | SYMBOL_SUBAREA |
| Displacement | 3 | SYMBOL_DISPLACEMENT |
| Index | 3 | SYMBOL_INDEX |
| Grantee on Area | 3 | GU_AREA |
| Area in File | 3 | FILE_AREA |
| DMCL in File | 3 | FILE_DMCL |
| Segment in DBName | 3 | DBNAME_SEGMENT |
| Subschema Mapping in DBName | 3 | DBNAME_SSCMAP |
| Schema referencing DBName | 3 | DBNAME_SCHEMA |
| Grantee for DBName | 3 | GU_DBNAME |
| Component of Loadlist | 3 | LOADLIST_COMPONENT |
| Pterm | 3 | PTERM |
| Lterm in Line | 3 | LTERM |
| Lterm in Destination | 3 | DESTLTERM |
| Printer in Destination | 3 | PRINTER |
| User in Destination | 3 | DESTINATION_DICTUSER |
| Entry of Map Table | 3 | MAPTYPE_ENTRY |
| Application | 3 | APPLICATION |
| Element Synonym | 4 | ELEMSYN |
| Element Description | 4 | ELEMSDES |
| Element Value | 4 | ELEMVALUE |
| Element Indexed By | 4 | NAMEDES |
| Record Procedure | 4 | MBR_RECORDPROC |
| Data Compression Table | 4 | MBR_DATACOMPTAB |
| Record Synonym | 4 | MBR_RECORDSYN |

Table 1 — Objectnames in Tree Order, Level and Objectid

| Objectname | Level | Objectid |
|--------------------------------|-------|--------------------------|
| Record Element | 4 | MBR_RECORDELEM |
| Set owned by Record | 4 | MBR_RECORDTYP_SET_OWNING |
| Set Record is member | 4 | MBR_RECORDTYP_SET_MEMBER |
| Record Control Key | 4 | SET_REC_CONTROLKEY |
| Record Foreign Key | 4 | SET_REC_FOREIGNKEY |
| Element of Subschema Record | 4 | SUBRECORDELEM |
| Area used in Program | 4 | SUBPROGAREA |
| Record used in Program | 4 | SUBPROGRECORD |
| Set used in Program | 4 | SUBPROGSET |
| Lterm in Pterm | 4 | PTERM_LTERM |
| Area Statistics for Program | 5 | AREASTATISTICS |
| Schema definition of Area | 5 | SUBPROGAREA_NSQAREA |
| Record Statistics for Program | 5 | RECORDSTATISTICS |
| Subschema definition of Record | 5 | SUBPROGRECORD_SUBRECORD |
| Set Statistics for Program | 5 | SETSTATISTICS |
| Schema definition of Set | 5 | SUBPROGSET_NSQSET |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|-----------------------------------|-------|----------------|
| Access Module | 2 | AM |
| Access Module for Table | 3 | TABLE_AM |
| Access Module for Table Procedure | 3 | PROCEDURE_AM |
| Access Module for View | 3 | VIEW_AM |
| Access Module granted | 2 | GP_GR_AM |
| Access Module granted | 2 | GP_US_AM |
| Access Module in Category | 2 | CATEGORY_AM |
| Activity | 1 | ACTIVITY |
| Activity granted | 2 | GP_GR_ACTIVITY |
| Activity granted | 2 | GP_US_ACTIVITY |
| Application | 3 | APPLICATION |
| Archive Journal | 2 | JOURNAL_ARCH |
| Area | 2 | AREA |
| Area Procedure | 3 | AREAPROC |
| Area Statistics for Program | 5 | AREASTATISTICS |
| Area granted | 2 | GP_GR_AREA |
| Area granted | 2 | GP_US_AREA |
| Area in DMCL | 2 | DMCL_AREA |
| Area in File | 3 | FILE_AREA |
| Area in Non SQL Schema | 2 | NSQLAREA |
| Area in Subschema | 3 | SUBSCHEMA_AREA |
| Area used in Program | 4 | SUBPROGAREA |
| Attribute | 2 | ATTRIB |
| Attribute | 2 | ATTRIBUTE |
| Attribute | 2 | SYSATTRIB |
| Autotask | 2 | AUTOTASK |
| Calc Key | 3 | CALC |
| Category | 1 | CATEGORY |
| Category granted | 2 | GP_GR_CATEGORY |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|------------------------------|-------|--------------------|
| Category granted | 2 | GP_US_CATEGORY |
| Central User | 1 | USER |
| Check Condition | 3 | CHECKCONDITION |
| Class / Attribute | 2 | NSQLATTRIB |
| Class / Attribute | 3 | SUBSCHEMAATTRIB |
| Column | 3 | COLUMN |
| Column of View | 3 | VIEWCOLUMN |
| Comment | 2 | NSQLCOMMENT |
| Comment | 3 | SUBSCHEMACOMMENT |
| Component of Loadlist | 3 | LOADLIST_COMPONENT |
| Constraint | 2 | CONSTRAINT |
| Constraint Table referenced | 3 | TABLE_CONST_REFCD |
| Constraint Table referencing | 3 | TABLE_CONST_REFING |
| DBADMIN granted | 2 | GP_GR_DBADMIN |
| DBADMIN granted | 2 | GP_US_DBADMIN |
| DBGroup in DBTable | 2 | DBGROUP |
| DBName | 2 | DBNAME |
| DBName & Segment granted | 2 | GP_GR_DBNAME |
| DBName & Segment granted | 2 | GP_US_DBNAME |
| DBName in Segment | 2 | SEGMENT_DBNAME |
| DBTable | 1 | DBTABLE |
| DBTable granted | 2 | GP_GR_DBTABLE |
| DBTable granted | 2 | GP_US_DBTABLE |
| DCADMIN granted | 2 | GP_GR_DCADMIN |
| DCADMIN granted | 2 | GP_US_DCADMIN |
| DMCL | 1 | DMCL |
| DMCL granted | 2 | GP_GR_DMCL |
| DMCL granted | 2 | GP_US_DMCL |
| DMCL in DBTable | 2 | DBTABLE_DMCL |
| DMCL in File | 3 | FILE_DMCL |
| DMCL including Segment | 2 | SEGMENT_DMCL |
| DMCL using Area | 3 | AREA_DMCL |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|----------------------------------|-------|---------------------|
| Data Compression Table | 3 | DATACOMPTAB |
| Data Compression Table | 4 | MBR_DATACOMPTAB |
| Database Buffer | 2 | BUFFER_DB |
| Destination | 2 | DESTINATION |
| Destination Node in Res Table | 2 | RESTABLE_DESTNODE |
| Dictionary | 0 | DATABASE |
| Dictionary User | 1 | DICTUSER |
| Dictionary via Node in Res Table | 2 | RESTABLE_DBNAME_VIA |
| Disk Journal | 2 | JOURNAL_DISK |
| Displacement | 3 | SYMBOL_DISPLACEMENT |
| Element Description | 4 | ELEMSDES |
| Element Indexed By | 4 | NAMEDES |
| Element Synonym | 4 | ELEMSYN |
| Element Value | 4 | ELEMVALUE |
| Element of Subschema Record | 4 | SUBRECORDELEM |
| Entity Type Authority | 2 | GP_DU |
| Entry of Map Table | 3 | MAPTYPE_ENTRY |
| File | 2 | FILE |
| File in Area | 3 | AREA_FILE |
| File in DMCL | 2 | DMCL_FILE |
| Grantee for DBName | 3 | GU_DBNAME |
| Grantee for DBTable | 2 | GU_DBTABLE |
| Grantee for DCADMIN | 1 | GU_DCADMIN |
| Grantee for DMCL | 2 | GU_DMCL |
| Grantee for Execute | 2 | GU_ACTIVITY |
| Grantee for Execute | 2 | GU_CATEGORY |
| Grantee for SYSADMIN | 1 | GU_SYSADMIN |
| Grantee for Schema | 2 | GU_SCHEMA |
| Grantee for Segment | 2 | GU_SEGMENT |
| Grantee for System Id | 2 | GU_SYSTEMID |
| Grantee for Use | 2 | GU_NSQLSHEMA |
| Grantee on Access Module | 3 | GU_AM |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|-------------------------------|-------|-----------------------|
| Grantee on Area | 3 | GU_AREA |
| Grantee on Group | 2 | GU_GROUP |
| Grantee on System Profile | 2 | GU_SYSPROFILE |
| Grantee on Table | 3 | GU_TABLE |
| Grantee on Table Procedure | 3 | GU_PROCEDURE |
| Grantee on User | 2 | GU_USER |
| Grantee on User Profile | 2 | GU_PROFILE |
| Grantee on View | 3 | GU_VIEW |
| Group | 1 | GROUP |
| Group User belongs | 2 | USER_GROUP |
| Group granted | 2 | GP_GR_GROUP |
| Group granted | 2 | GP_US_GROUP |
| IDD Class | 1 | CLASS |
| IDD Element Description | 3 | IDDELEMSDES |
| IDD Element Indexed By | 3 | IDDNAMEDES |
| IDD Element Synonym | 3 | IDDELEMSYN |
| IDD Element Value | 3 | IDDELEMVALUE |
| IDD Record | 1 | IDDRECORD |
| IDD Record Element | 2 | IDDRECORDELEM |
| IDD Record Synonym | 2 | IDDRECORDSYN |
| Index | 3 | INDEX |
| Index | 3 | SYMBOL_INDEX |
| Index stored in Area | 3 | AREA_INDEX |
| Journal Buffer | 2 | BUFFER_JOURNAL |
| Key Table | 2 | KEYTABLE |
| Key in Table Procedure | 3 | PROCKEY |
| Line | 2 | LINE |
| Load List | 2 | LOADLIST |
| Load Module | 3 | LOADMODULE |
| Load Module in Category | 2 | CATEGORY_LOADMODULE |
| Local Dictionary in Res Table | 2 | RESTABLE_DBNAME_LOCAL |
| Lterm | 2 | SYSTEM_LTERM |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|-------------------------------|-------|--------------------|
| Lterm in Destination | 3 | DESTLTERM |
| Lterm in Line | 3 | LTERM |
| Lterm in Pterm | 4 | PTERM_LTERM |
| Map Table | 2 | MAPTYPE |
| Member Record of Set | 3 | SET_RECORD_MEMBER |
| Network Record accessed | 3 | AM_RECORDTYP |
| Node | 2 | NODE |
| Non SQL Schema | 1 | NSQLSCHEMA |
| Non SQL Schema granted | 2 | GP_GR_NSQLSCHEMA |
| Non SQL Schema granted | 2 | GP_US_NSQLSCHEMA |
| Non SQL Schema registered | 2 | RP_DU_NSQLSCHEMA |
| Non SQL Schema responsibility | 2 | SP_DU_NSQLSCHEMA |
| Owner Record of Set | 3 | SET_RECORD_OWNER |
| Parameter of Table Procedure | 3 | PROCCOLUMN |
| Printer | 2 | SYSTEM_PRINTER |
| Printer in Destination | 3 | PRINTER |
| Program | 2 | PROGRAM |
| Program in Category | 2 | CATEGORY_PROGRAM |
| Program using Subschema | 3 | SUBSCHEMA_PROGRAM |
| Pterm | 3 | PTERM |
| Queue | 2 | QUEUE |
| Queue in Category | 2 | CATEGORY_QUEUE |
| Record | 2 | RECORDTYP |
| Record Control Key | 4 | SET_REC_CONTROLKEY |
| Record Element | 3 | RECORDELEM |
| Record Element | 4 | MBR_RECORDELEM |
| Record Foreign Key | 4 | SET_REC_FOREIGNKEY |
| Record Procedure | 3 | RECORDPROC |
| Record Procedure | 4 | MBR_RECORDPROC |
| Record Statistics for Program | 5 | RECORDSTATISTICS |
| Record Synonym | 3 | RECORDSYN |
| Record Synonym | 4 | MBR_RECORDSYN |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|---------------------------------|-------|--------------------------|
| Record in Area | 3 | AREA_RECORDTYP |
| Record in Subschema | 3 | SUBRECORD |
| Record used in Program | 4 | SUBPROGRECORD |
| Referenced Table in Constraint | 3 | CONST_TABLE_REFCED |
| Referencing Table in Constraint | 3 | CONST_TABLE_REFING |
| Registree for Schema | 2 | RU_NSQLSCHEMA |
| Registree for Subschema | 3 | RU_SUBSCHEMA |
| Relational Command Module | 1 | RCM |
| Responsible for Schema | 2 | SU_NSQLSCHEMA |
| Responsible for Subschema | 3 | SU_SUBSCHEMA |
| Rununit | 2 | RUNUNIT |
| Rununit in Category | 2 | CATEGORY_RUNUNIT |
| SQL Schema | 1 | SCHEMA |
| SYSADMIN granted | 2 | GP_GR_SYSADMIN |
| SYSADMIN granted | 2 | GP_US_SYSADMIN |
| Schema definition of Area | 5 | SUBPROGAREA_NSQAREA |
| Schema definition of Set | 5 | SUBPROGSET_NSQSET |
| Schema granted | 2 | GP_GR_SCHEMA |
| Schema granted | 2 | GP_US_SCHEMA |
| Schema referencing DBName | 3 | DBNAME_SCHEMA |
| Schema referencing Segment | 2 | SEGMENT_SCHEMA |
| Schema using Area as default | 3 | AREA_SCHEMA |
| Segment | 1 | SEGMENT |
| Segment in DBName | 3 | DBNAME_SEGMENT |
| Segment in DMCL | 2 | DMCL_SEGMENT |
| Segment.Area Override | 2 | DMCL_SEGMENT_AREA_OVER |
| Segment.File Override | 2 | DMCL_SEGMENT_FILE_OVER |
| Set | 2 | NSQLSET |
| Set Record is member | 3 | RECORDTYP_SET_MEMBER |
| Set Record is member | 4 | MBR_RECORDTYP_SET_MEMBER |
| Set Statistics for Program | 5 | SETSTATISTICS |
| Set in Subschema | 3 | SUBSCHEMA_SET |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|--------------------------------|-------|--------------------------|
| Set owned by Record | 3 | RECORDTYP_SET_OWNING |
| Set owned by Record | 4 | MBR_RECORDTYP_SET_OWNING |
| Set used in Program | 4 | SUBPROGSET |
| Storage Pool | 2 | STORPOOL |
| Structure Shared By | 3 | RECORDSHARE |
| Subarea | 3 | SYMBOL_SUBAREA |
| Subschema | 2 | SUBSCHEMA |
| Subschema Mapping | 2 | SSCMAP |
| Subschema Mapping in DBName | 3 | DBNAME_SSCMAP |
| Subschema definition of Record | 5 | SUBPROGRECORD_SUBRECORD |
| Subschema registered | 2 | RP_DU_SUBSCHEMA |
| Subschema responsibility | 2 | SP_DU_SUBSCHEMA |
| System | 1 | SYSTEM |
| System Id | 1 | SYSTEMID |
| System Id granted | 2 | GP_GR_SYSTEMID |
| System Id granted | 2 | GP_US_SYSTEMID |
| System Profile | 1 | SYSPROFILE |
| System Profile granted | 2 | GP_GR_SYSPROFILE |
| System Profile granted | 2 | GP_US_SYSPROFILE |
| Table | 1 | ALL_TABLE |
| Table Procedure | 1 | ALL_PROCEDURE |
| Table Procedure accessed | 3 | AM_PROCEDURE |
| Table Procedure granted | 2 | GP_GR_PROCEDURE |
| Table Procedure granted | 2 | GP_US_PROCEDURE |
| Table Procedure in Schema | 2 | PROCEDURE |
| Table accessed | 3 | AM_TABLE |
| Table granted | 2 | GP_GR_TABLE |
| Table granted | 2 | GP_US_TABLE |
| Table in Schema | 2 | TABLE |
| Table stored in Area | 3 | AREA_TABLE |
| Tape Journal | 2 | JOURNAL_TAPE |
| Task | 2 | TASK |

Table 2 — Objectnames in Alphabetic Order, Level and Objectid

| Objectname | Level | Objectid |
|----------------------|-------|----------------------|
| Task in Category | 2 | CATEGORY_TASK |
| User Profile | 1 | PROFILE |
| User Profile granted | 2 | GP_GR_PROFILE |
| User Profile granted | 2 | GP_US_PROFILE |
| User granted | 2 | GP_GR_USER |
| User granted | 2 | GP_US_USER |
| User in Destination | 3 | DESTINATION_DICTUSER |
| User in Group | 2 | GROUP_USER |
| View | 1 | ALL_VIEW |
| View Component | 3 | VIEW_COMPONENT |
| View Definition | 3 | VIEWDEFINITION |
| View accessed | 3 | AM_VIEW |
| View granted | 2 | GP_GR_VIEW |
| View granted | 2 | GP_US_VIEW |
| View in Schema | 2 | VIEW |
| View on Table | 3 | TABLE_VIEW |
| XA Storage Pool | 2 | XASTORPOOL |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|-----------------------------------|-------|-----------------------------|
| Access Module | 2 | SYSTEM.AM |
| Access Module for Table | 3 | SYSVSYST.AM_AMDEP |
| Access Module for Table Procedure | 3 | SYSVSYST.AM_AMDEP |
| Access Module for View | 3 | SYSVSYST.AM_AMDEP |
| Access Module granted | 2 | SYSTEM.RESOURCEAUTH |
| Access Module granted | 2 | SYSTEM.RESOURCEAUTH |
| Access Module in Category | 2 | IDMSSECS.RESOURCE |
| Activity | 1 | IDMSSECS.RESOURCE |
| Activity granted | 2 | IDMSSECS.RESGROUPAUTH |
| Activity granted | 2 | IDMSSECS.RESGROUPAUTH |
| Application | 3 | SYSVNTWK.APPLICATION |
| Archive Journal | 2 | SYSTEM.JOURNAL |
| Area | 2 | SYSTEM.AREA |
| Area Procedure | 3 | SYSVNTWK.AREAPROC |
| Area Statistics for Program | 5 | SYSVNTWK.SUBPROGAREA_NSQAR |
| Area granted | 2 | IDMSSECS.RESOURCEAUTH |
| Area granted | 2 | IDMSSECS.RESOURCEAUTH |
| Area in DMCL | 2 | SYSVSYST.DMCL_AREA |
| Area in File | 3 | SYSTEM.FILEMAP |
| Area in Non SQL Schema | 2 | SYSVNTWK.NSQLAREA |
| Area in Subschema | 3 | SYSVNTWK.SUBSCHEMA_AREA |
| Area used in Program | 4 | SYSVNTWK.SUBPROGAREA |
| Attribute | 2 | IDMSSECS.ATTRIBUTE |
| Attribute | 2 | IDMSSECU.ATTRIBUTE |
| Attribute | 2 | SYSVNTWK.ATTRIBUTE |
| Autotask | 2 | SYSVNTWK.AUTOTASK |
| Calc Key | 3 | SYSTEM.INDEX |
| Category | 1 | IDMSSECS.RESOURCEGROUP |
| Category granted | 2 | IDMSSECS.RESGROUPAUTH |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|------------------------------|-------|-----------------------------|
| Category granted | 2 | IDMSSECS.RESGROUPAUTH |
| Central User | 1 | IDMSSECU.USER |
| Check Condition | 3 | SYSTEM.SYNTAX |
| Class / Attribute | 2 | SYSVNTWK.NSQLATTRIB |
| Class / Attribute | 3 | SYSVNTWK.SUBSCHEMAATTRIB |
| Column | 3 | SYSTEM.COLUMN |
| Column of View | 3 | SYSTEM.COLUMN |
| Comment | 2 | SYSVNTWK.NSQLCOMMENT |
| Comment | 3 | SYSVNTWK.SUBSCHEMACOMMENT |
| Component of Loadlist | 3 | SYSVNTWK.LOADLIST_COMPONENT |
| Constraint | 2 | SYSVSYST.CONSTRAINT |
| Constraint Table referenced | 3 | SYSVSYST.CONSTRAINT |
| Constraint Table referencing | 3 | SYSVSYST.CONSTRAINT |
| DBADMIN granted | 2 | IDMSSECS.RESOURCEAUTH |
| DBADMIN granted | 2 | IDMSSECS.RESOURCEAUTH |
| DBGroup in DBTable | 2 | SYSTEM.DBNAME |
| DBName | 2 | SYSTEM.DBNAME |
| DBName & Segment granted | 2 | IDMSSECS.RESOURCEAUTH |
| DBName & Segment granted | 2 | IDMSSECS.RESOURCEAUTH |
| DBName in Segment | 2 | SYSTEM.DBSEGMENT |
| DBTable | 1 | SYSTEM.DBTABLE |
| DBTable granted | 2 | IDMSSECS.RESOURCEAUTH |
| DBTable granted | 2 | IDMSSECS.RESOURCEAUTH |
| DCADMIN granted | 2 | IDMSSECS.RESOURCEAUTH |
| DCADMIN granted | 2 | IDMSSECS.RESOURCEAUTH |
| DMCL | 1 | SYSTEM.DMCL |
| DMCL granted | 2 | IDMSSECS.RESOURCEAUTH |
| DMCL granted | 2 | IDMSSECS.RESOURCEAUTH |
| DMCL in DBTable | 2 | SYSTEM.DMCL |
| DMCL in File | 3 | SYSVSYST.DMCL_FILE |
| DMCL including Segment | 2 | SYSTEM.DMCLSEGMENT |
| DMCL using Area | 3 | SYSVSYST.DMCL_AREA |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|----------------------------------|-------|--|
| Data Compression Table | 3 | SYSVNTWK.RECORDPROC |
| Data Compression Table | 4 | SYSVNTWK.RECORDPROC, SYSTSCHM.SOR-046 |
| Database Buffer | 2 | SYSTEM.BUFFER |
| Destination | 2 | SYSVNTWK.DESTINATION |
| Destination Node in Res Table | 2 | SYSVNTWK.RESTABLE |
| Dictionary | 0 | SYSTEM.DBNAME |
| Dictionary User | 1 | SYSVNTWK.DICTUSER |
| Dictionary via Node in Res Table | 2 | SYSVNTWK.RESTABLE |
| Disk Journal | 2 | SYSTEM.JOURNAL |
| Displacement | 3 | SYSTEM.SYMBOL |
| Element Description | 4 | SYSVNTWK.ELEMSDES |
| Element Indexed By | 4 | SYSVNTWK.NAMEDES |
| Element Synonym | 4 | SYSVNTWK.ELEMSYN |
| Element Value | 4 | SYSVNTWK.ELEMVALUE |
| Element of Subschema Record | 4 | SYSVNTWK.SUBRECORDELEM |
| Entity Type Authority | 2 | SYSVNTWK.DICTUSER |
| Entry of Map Table | 3 | SYSVNTWK.MAPTYPE |
| File | 2 | SYSTEM.FILE |
| File in Area | 3 | SYSTEM.FILEMAP |
| File in DMCL | 2 | SYSVSYST.DMCL_FILE |
| Grantee for DBName | 3 | IDMSSECS.RESOURCEAUTH |
| Grantee for DBTable | 2 | IDMSSECS.RESOURCEAUTH |
| Grantee for DCADMIN | 1 | IDMSSECS.RESOURCEAUTH |
| Grantee for DMCL | 2 | IDMSSECS.RESOURCEAUTH |
| Grantee for Execute | 2 | IDMSSECS.RESGROUPAUTH |
| Grantee for Execute | 2 | IDMSSECS.RESGROUPAUTH |
| Grantee for SYSADMIN | 1 | IDMSSECU.RESOURCEAUTH |
| Grantee for Schema | 2 | SYSTEM.RESOURCEAUTH |
| Grantee for Segment | 2 | IDMSSECS.RESOURCEAUTH |
| Grantee for System Id | 2 | IDMSSECS.RESOURCEAUTH |
| Grantee for Use | 2 | IDMSSECS.RESOURCEAUTH |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|-------------------------------|-------|-----------------------------|
| Grantee on Access Module | 3 | SYSTEM.RESOURCEAUTH |
| Grantee on Area | 3 | IDMSSECS.RESOURCEAUTH |
| Grantee on Group | 2 | IDMSSECU.RESOURCEAUTH |
| Grantee on System Profile | 2 | IDMSSECS.RESOURCEAUTH |
| Grantee on Table | 3 | SYSTEM.RESOURCEAUTH |
| Grantee on Table Procedure | 3 | SYSTEM.RESOURCEAUTH |
| Grantee on User | 2 | IDMSSECU.RESOURCEAUTH |
| Grantee on User Profile | 2 | IDMSSECU.RESOURCEAUTH |
| Grantee on View | 3 | SYSTEM.RESOURCEAUTH |
| Group | 1 | IDMSSECU.USER |
| Group User belongs | 2 | IDMSSECU.USERGROUP |
| Group granted | 2 | IDMSSECU.RESOURCEAUTH |
| Group granted | 2 | IDMSSECU.RESOURCEAUTH |
| IDD Class | 1 | SYSVNTWK.CLASS |
| IDD Element Description | 3 | SYSVNTWK.ELEMSDES |
| IDD Element Indexed By | 3 | SYSVNTWK.NAMEDES |
| IDD Element Synonym | 3 | SYSVNTWK.IDDELEMSYN |
| IDD Element Value | 3 | SYSVNTWK.ELEMVALUE |
| IDD Record | 1 | SYSVNTWK.IDDRECORD |
| IDD Record Element | 2 | SYSVNTWK.RECORDELEM |
| IDD Record Synonym | 2 | SYSVNTWK.IDDRECORDSYN |
| Index | 3 | SYSTEM.INDEX |
| Index | 3 | SYSTEM.SYMBOL |
| Index stored in Area | 3 | SYSTEM.INDEX |
| Journal Buffer | 2 | SYSTEM.BUFFER |
| Key Table | 2 | SYSVNTWK.KEYTABLE |
| Key in Table Procedure | 3 | SYSTEM.INDEX |
| Line | 2 | SYSVNTWK.LINE |
| Load List | 2 | SYSVNTWK.LOADLIST |
| Load Module | 3 | SYSVNTWK.LOADMODULE |
| Load Module in Category | 2 | IDMSSECS.RESOURCE |
| Local Dictionary in Res Table | 2 | SYSVNTWK.RESTABLE |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|-------------------------------|-------|--|
| Lterm | 2 | SYSVNTWK.SYSTEM_LTERM |
| Lterm in Destination | 3 | SYSVNTWK.DESTLTERM |
| Lterm in Line | 3 | SYSVNTWK.PTERM_LTERM |
| Lterm in Pterm | 4 | SYSVNTWK.PTERM_LTERM |
| Map Table | 2 | SYSVNTWK.MAPTYPE |
| Member Record of Set | 3 | SYSVNTWK.SET_RECORD_MEMBER |
| Network Record accessed | 3 | SYSVNTWK.RECORDTYP |
| Node | 2 | SYSVNTWK.NODE |
| Non SQL Schema | 1 | SYSVNTWK.NSQLSCHEMA |
| Non SQL Schema granted | 2 | IDMSSECS.RESOURCEAUTH |
| Non SQL Schema granted | 2 | IDMSSECS.RESOURCEAUTH |
| Non SQL Schema registered | 2 | SYSVNTWK.U_NSQLSCHEMA |
| Non SQL Schema responsibility | 2 | SYSVNTWK.U_NSQLSCHEMA |
| Owner Record of Set | 3 | SYSVNTWK.SET_RECORD_OWNER |
| Parameter of Table Procedure | 3 | SYSTEM.COLUMN |
| Printer | 2 | SYSVNTWK.SYSTEM_PRINTER |
| Printer in Destination | 3 | SYSVNTWK.PRINTER |
| Program | 2 | SYSVNTWK.PROGRAM |
| Program in Category | 2 | IDMSSECS.RESOURCE |
| Program using Subschema | 3 | SYSVNTWK.SUBSCHEMA_PROG |
| Pterm | 3 | SYSVNTWK.PTERM |
| Queue | 2 | SYSVNTWK.QUEUE |
| Queue in Category | 2 | IDMSSECS.RESOURCE |
| Record | 2 | SYSVNTWK.RECORDTYP |
| Record Control Key | 4 | SYSVNTWK.SET_REC_CONTROLKEY |
| Record Element | 3 | SYSVNTWK.RECORDELEM |
| Record Element | 4 | SYSVNTWK.RECORDELEM, SYSTSCHM.SOR-046 |
| Record Foreign Key | 4 | SYSVNTWK.SET_REC_FOREIGNKEY |
| Record Procedure | 3 | SYSVNTWK.RECORDPROC |
| Record Procedure | 4 | SYSVNTWK.RECORDPROC, SYSTSCHM.SOR-046 |
| Record Statistics for Program | 5 | SYSVNTWK.SUBPROGRECORD_SUBR |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|---------------------------------|-------|---|
| Record Synonym | 3 | SYSVNTWK.RECORDSYN |
| Record Synonym | 4 | SYSVNTWK.RECORDSYN, SYSTSCHM.SOR-046 |
| Record in Area | 3 | SYSVNTWK.RECORDTYP |
| Record in Subschema | 3 | SYSVNTWK.SUBRECORD |
| Record used in Program | 4 | SYSVNTWK.SUBPROGRECORD |
| Referenced Table in Constraint | 3 | SYSVSYST.CONSTRAINT |
| Referencing Table in Constraint | 3 | SYSVSYST.CONSTRAINT |
| Registree for Schema | 2 | SYSVNTWK.U_NSQLSHEMA |
| Registree for Subschema | 3 | SYSVNTWK.U_SUBSCHEMA |
| Relational Command Module | 1 | SYSVNTWK.RCM |
| Responsible for Schema | 2 | SYSVNTWK.U_NSQLSHEMA |
| Responsible for Subschema | 3 | SYSVNTWK.U_SUBSCHEMA |
| Rununit | 2 | SYSVNTWK.RUNUNIT |
| Rununit in Category | 2 | IDMSSECS.RESOURCE |
| SQL Schema | 1 | SYSTEM.SCHEMA |
| SYSADMIN granted | 2 | IDMSSECU.RESOURCEAUTH |
| SYSADMIN granted | 2 | IDMSSECU.RESOURCEAUTH |
| Schema definition of Area | 5 | SYSVNTWK.SUBPROGAREA_NSQLAR |
| Schema definition of Set | 5 | SYSVNTWK.SUBPROGSET_NSQSET |
| Schema granted | 2 | SYSTEM.RESOURCEAUTH |
| Schema granted | 2 | SYSTEM.RESOURCEAUTH |
| Schema referencing DBName | 3 | SYSVSYST.DBNAME_SCHEMA |
| Schema referencing Segment | 2 | SYSTEM.SCHEMA |
| Schema using Area as default | 3 | SYSTEM.SCHEMA |
| Segment | 1 | SYSTEM.SEGMENT |
| Segment in DBName | 3 | SYSTEM.DBSEGMENT |
| Segment in DMCL | 2 | SYSTEM.DMCLSEGMENT |
| Segment.Area Override | 2 | SYSVSYST.VDMCLAREA |
| Segment.File Override | 2 | SYSVSYST.VDMCLFILE |
| Set | 2 | SYSVNTWK.NSQLSET |
| Set Record is member | 3 | SYSVNTWK.RECORDTYP_SET_ME |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|--------------------------------|-------|-----------------------------|
| Set Record is member | 4 | SYSVNTWK.MBR_REC_SET_ME |
| Set Statistics for Program | 5 | SYSVNTWK.SUBPROGSET_NSQSET |
| Set in Subschema | 3 | SYSVNTWK.SUBSCHEMA_SET |
| Set owned by Record | 3 | SYSVNTWK.RECORDTYP_SET_OW |
| Set owned by Record | 4 | SYSVNTWK.MBR_REC_SET_OW |
| Set used in Program | 4 | SYSVNTWK.SUBPROGSET |
| Storage Pool | 2 | SYSVNTWK.STORPOOL |
| Structure Shared By | 3 | SYSVNTWK.RECORDSHARE |
| Subarea | 3 | SYSTEM.SYMBOL |
| Subschema | 2 | SYSVNTWK.SUBSCHEMA |
| Subschema Mapping | 2 | SYSTEM.DBSSC |
| Subschema Mapping in DBName | 3 | SYSTEM.DBSSC |
| Subschema definition of Record | 5 | SYSVNTWK.SUBPROGRECORD_SUBR |
| Subschema registered | 2 | SYSVNTWK.U_SUBSCHEMA |
| Subschema responsibility | 2 | SYSVNTWK.U_SUBSCHEMA |
| System | 1 | SYSVNTWK.SYSTEM |
| System Id | 1 | IDMSSECS.RESOURCE |
| System Id granted | 2 | IDMSSECS.RESOURCEAUTH |
| System Id granted | 2 | IDMSSECS.RESOURCEAUTH |
| System Profile | 1 | IDMSSECS.PROFILE |
| System Profile granted | 2 | IDMSSECS.RESOURCEAUTH |
| System Profile granted | 2 | IDMSSECS.RESOURCEAUTH |
| Table | 1 | SYSTEM.TABLE |
| Table Procedure | 1 | SYSTEM.TABLE |
| Table Procedure accessed | 3 | SYSVSYST.AM_AMDEP |
| Table Procedure granted | 2 | SYSTEM.RESOURCEAUTH |
| Table Procedure granted | 2 | SYSTEM.RESOURCEAUTH |
| Table Procedure in Schema | 2 | SYSTEM.TABLE |
| Table accessed | 3 | SYSVSYST.AM_AMDEP |
| Table granted | 2 | SYSTEM.RESOURCEAUTH |
| Table granted | 2 | SYSTEM.RESOURCEAUTH |
| Table in Schema | 2 | SYSTEM.TABLE |

Table 3 — Objectnames in Alphabetic Order, Level and Resourcenames

| Objectnames | Level | Schema.Table used by Object |
|----------------------|--------------|------------------------------------|
| Table stored in Area | 3 | SYSTEM.TABLE |
| Tape Journal | 2 | SYSTEM.JOURNAL |
| Task | 2 | SYSVNTWK.TASK |
| Task in Category | 2 | IDMSSECS.RESOURCE |
| User Profile | 1 | IDMSSECU.PROFILE |
| User Profile granted | 2 | IDMSSECU.RESOURCEAUTH |
| User Profile granted | 2 | IDMSSECU.RESOURCEAUTH |
| User granted | 2 | IDMSSECU.RESOURCEAUTH |
| User granted | 2 | IDMSSECU.RESOURCEAUTH |
| User in Destination | 3 | SYSVNTWK.DESTINATION_DICTUS |
| User in Group | 2 | IDMSSECU.USERGROUP |
| View | 1 | SYSTEM.TABLE |
| View Component | 3 | SYSTEM.VIEWDEP |
| View Definition | 3 | SYSTEM.SYNTAX |
| View accessed | 3 | SYSVSYST.AM_AMDEP |
| View granted | 2 | SYSTEM.RESOURCEAUTH |
| View granted | 2 | SYSTEM.RESOURCEAUTH |
| View in Schema | 2 | SYSTEM.TABLE |
| View on Table | 3 | SYSTEM.VIEWDEP |
| XA Storage Pool | 2 | SYSVNTWK.STORPOOL |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|-----------------------------|-------|---------------------------|
| IDMSSECS.ATTRIBUTE | 2 | Attribute |
| IDMSSECS.PROFILE | 1 | System Profile |
| IDMSSECS.RESGROUPAUTH | 2 | Activity granted |
| IDMSSECS.RESGROUPAUTH | 2 | Activity granted |
| IDMSSECS.RESGROUPAUTH | 2 | Category granted |
| IDMSSECS.RESGROUPAUTH | 2 | Category granted |
| IDMSSECS.RESGROUPAUTH | 2 | Grantee for Execute |
| IDMSSECS.RESGROUPAUTH | 2 | Grantee for Execute |
| IDMSSECS.RESOURCE | 1 | Activity |
| IDMSSECS.RESOURCE | 1 | System Id |
| IDMSSECS.RESOURCE | 2 | Access Module in Category |
| IDMSSECS.RESOURCE | 2 | Load Module in Category |
| IDMSSECS.RESOURCE | 2 | Program in Category |
| IDMSSECS.RESOURCE | 2 | Queue in Category |
| IDMSSECS.RESOURCE | 2 | Rununit in Category |
| IDMSSECS.RESOURCE | 2 | Task in Category |
| IDMSSECS.RESOURCEAUTH | 1 | Grantee for DCADMIN |
| IDMSSECS.RESOURCEAUTH | 2 | Area granted |
| IDMSSECS.RESOURCEAUTH | 2 | Area granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBADMIN granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBADMIN granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBName & Segment granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBName & Segment granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBTable granted |
| IDMSSECS.RESOURCEAUTH | 2 | DBTable granted |
| IDMSSECS.RESOURCEAUTH | 2 | DCADMIN granted |
| IDMSSECS.RESOURCEAUTH | 2 | DCADMIN granted |
| IDMSSECS.RESOURCEAUTH | 2 | DMCL granted |
| IDMSSECS.RESOURCEAUTH | 2 | DMCL granted |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|---------------------------|
| IDMSSECS.RESOURCEAUTH | 2 | Grantee for DBTable |
| IDMSSECS.RESOURCEAUTH | 2 | Grantee for DMCL |
| IDMSSECS.RESOURCEAUTH | 2 | Grantee for Segment |
| IDMSSECS.RESOURCEAUTH | 2 | Grantee for System Id |
| IDMSSECS.RESOURCEAUTH | 2 | Grantee for Use |
| IDMSSECS.RESOURCEAUTH | 2 | Grantee on System Profile |
| IDMSSECS.RESOURCEAUTH | 2 | Non SQL Schema granted |
| IDMSSECS.RESOURCEAUTH | 2 | Non SQL Schema granted |
| IDMSSECS.RESOURCEAUTH | 2 | System Id granted |
| IDMSSECS.RESOURCEAUTH | 2 | System Id granted |
| IDMSSECS.RESOURCEAUTH | 2 | System Profile granted |
| IDMSSECS.RESOURCEAUTH | 2 | System Profile granted |
| IDMSSECS.RESOURCEAUTH | 3 | Grantee for DBName |
| IDMSSECS.RESOURCEAUTH | 3 | Grantee on Area |
| IDMSSECS.RESOURCEGROUP | 1 | Category |
| IDMSSECU.ATTRIBUTE | 2 | Attribute |
| IDMSSECU.PROFILE | 1 | User Profile |
| IDMSSECU.RESOURCEAUTH | 1 | Grantee for SYSADMIN |
| IDMSSECU.RESOURCEAUTH | 2 | Grantee on Group |
| IDMSSECU.RESOURCEAUTH | 2 | Grantee on User |
| IDMSSECU.RESOURCEAUTH | 2 | Grantee on User Profile |
| IDMSSECU.RESOURCEAUTH | 2 | Group granted |
| IDMSSECU.RESOURCEAUTH | 2 | Group granted |
| IDMSSECU.RESOURCEAUTH | 2 | SYSADMIN granted |
| IDMSSECU.RESOURCEAUTH | 2 | SYSADMIN granted |
| IDMSSECU.RESOURCEAUTH | 2 | User Profile granted |
| IDMSSECU.RESOURCEAUTH | 2 | User Profile granted |
| IDMSSECU.RESOURCEAUTH | 2 | User granted |
| IDMSSECU.RESOURCEAUTH | 2 | User granted |
| IDMSSECU.USER | 1 | Central User |
| IDMSSECU.USER | 1 | Group |
| IDMSSECU.USERGROUP | 2 | Group User belongs |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|------------------------------|
| IDMSSECU.USERGROUP | 2 | User in Group |
| SYSTEM.AM | 2 | Access Module |
| SYSTEM.AREA | 2 | Area |
| SYSTEM.BUFFER | 2 | Database Buffer |
| SYSTEM.BUFFER | 2 | Journal Buffer |
| SYSTEM.COLUMN | 3 | Column |
| SYSTEM.COLUMN | 3 | Column of View |
| SYSTEM.COLUMN | 3 | Parameter of Table Procedure |
| SYSTEM.DBNAME | 0 | Dictionary |
| SYSTEM.DBNAME | 2 | DBGroup in DBTable |
| SYSTEM.DBNAME | 2 | DBName |
| SYSTEM.DBSEGMENT | 2 | DBName in Segment |
| SYSTEM.DBSEGMENT | 3 | Segment in DBName |
| SYSTEM.DBSSC | 2 | Subschema Mapping |
| SYSTEM.DBSSC | 3 | Subschema Mapping in DBName |
| SYSTEM.DBTABLE | 1 | DBTable |
| SYSTEM.DMCL | 1 | DMCL |
| SYSTEM.DMCL | 2 | DMCL in DBTable |
| SYSTEM.DMCLSEGMENT | 2 | DMCL including Segment |
| SYSTEM.DMCLSEGMENT | 2 | Segment in DMCL |
| SYSTEM.FILE | 2 | File |
| SYSTEM.FILEMAP | 3 | Area in File |
| SYSTEM.FILEMAP | 3 | File in Area |
| SYSTEM.INDEX | 3 | Calc Key |
| SYSTEM.INDEX | 3 | Index |
| SYSTEM.INDEX | 3 | Index stored in Area |
| SYSTEM.INDEX | 3 | Key in Table Procedure |
| SYSTEM.JOURNAL | 2 | Archive Journal |
| SYSTEM.JOURNAL | 2 | Disk Journal |
| SYSTEM.JOURNAL | 2 | Tape Journal |
| SYSTEM.RESOURCEAUTH | 2 | Access Module granted |
| SYSTEM.RESOURCEAUTH | 2 | Access Module granted |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|------------------------------|
| SYSTEM.RESOURCEAUTH | 2 | Grantee for Schema |
| SYSTEM.RESOURCEAUTH | 2 | Schema granted |
| SYSTEM.RESOURCEAUTH | 2 | Schema granted |
| SYSTEM.RESOURCEAUTH | 2 | Table Procedure granted |
| SYSTEM.RESOURCEAUTH | 2 | Table Procedure granted |
| SYSTEM.RESOURCEAUTH | 2 | Table granted |
| SYSTEM.RESOURCEAUTH | 2 | Table granted |
| SYSTEM.RESOURCEAUTH | 2 | View granted |
| SYSTEM.RESOURCEAUTH | 2 | View granted |
| SYSTEM.RESOURCEAUTH | 3 | Grantee on Access Module |
| SYSTEM.RESOURCEAUTH | 3 | Grantee on Table |
| SYSTEM.RESOURCEAUTH | 3 | Grantee on Table Procedure |
| SYSTEM.RESOURCEAUTH | 3 | Grantee on View |
| SYSTEM.SCHEMA | 1 | SQL Schema |
| SYSTEM.SCHEMA | 2 | Schema referencing Segment |
| SYSTEM.SCHEMA | 3 | Schema using Area as default |
| SYSTEM.SEGMENT | 1 | Segment |
| SYSTEM.SYMBOL | 3 | Displacement |
| SYSTEM.SYMBOL | 3 | Index |
| SYSTEM.SYMBOL | 3 | Subarea |
| SYSTEM.SYNTAX | 3 | Check Condition |
| SYSTEM.SYNTAX | 3 | View Definition |
| SYSTEM.TABLE | 1 | Table |
| SYSTEM.TABLE | 1 | Table Procedure |
| SYSTEM.TABLE | 1 | View |
| SYSTEM.TABLE | 2 | Table Procedure in Schema |
| SYSTEM.TABLE | 2 | Table in Schema |
| SYSTEM.TABLE | 2 | View in Schema |
| SYSTEM.TABLE | 3 | Table stored in Area |
| SYSTEM.VIEWDEP | 3 | View Component |
| SYSTEM.VIEWDEP | 3 | View on Table |
| SYSTSCHM.SOR-046 | 4 | Data Compression Table |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|-----------------------------|-------|-------------------------|
| SYSTSCHM.SOR-046 | 4 | Record Procedure |
| SYSTSCHM.SOR-046 | 4 | Record Synonym |
| SYSTSCHM.SOR-046 | 4 | Record Element |
| SYSVNTWK.APPLICATION | 3 | Application |
| SYSVNTWK.AREAPROC | 3 | Area Procedure |
| SYSVNTWK.ATTRIBUTE | 2 | Attribute |
| SYSVNTWK.AUTOTASK | 2 | Autotask |
| SYSVNTWK.CLASS | 1 | IDD Class |
| SYSVNTWK.DESTINATION | 2 | Destination |
| SYSVNTWK.DESTINATION_DICTUS | 3 | User in Destination |
| SYSVNTWK.DESTLTERM | 3 | Lterm in Destination |
| SYSVNTWK.DICTUSER | 1 | Dictionary User |
| SYSVNTWK.DICTUSER | 2 | Entity Type Authority |
| SYSVNTWK.ELEMSDES | 3 | IDD Element Description |
| SYSVNTWK.ELEMSDES | 4 | Element Description |
| SYSVNTWK.ELEMSYN | 4 | Element Synonym |
| SYSVNTWK.ELEMVALUE | 3 | IDD Element Value |
| SYSVNTWK.ELEMVALUE | 4 | Element Value |
| SYSVNTWK.IDDELEMSYN | 3 | IDD Element Synonym |
| SYSVNTWK.IDDRECORD | 1 | IDD Record |
| SYSVNTWK.IDDRECORDSYN | 2 | IDD Record Synonym |
| SYSVNTWK.KEYTABLE | 2 | Key Table |
| SYSVNTWK.LINE | 2 | Line |
| SYSVNTWK.LOADLIST | 2 | Load List |
| SYSVNTWK.LOADLIST_COMPONENT | 3 | Component of Loadlist |
| SYSVNTWK.LOADMODULE | 3 | Load Module |
| SYSVNTWK.MAPTYPE | 2 | Map Table |
| SYSVNTWK.MAPTYPE | 3 | Entry of Map Table |
| SYSVNTWK.MBR_REC_SET_ME | 4 | Set Record is member |
| SYSVNTWK.MBR_REC_SET_OW | 4 | Set owned by Record |
| SYSVNTWK.NAMEDES | 3 | IDD Element Indexed By |
| SYSVNTWK.NAMEDES | 4 | Element Indexed By |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|----------------------------------|
| SYSVNTWK.NODE | 2 | Node |
| SYSVNTWK.NSQLAREA | 2 | Area in Non SQL Schema |
| SYSVNTWK.NSQLATTRIB | 2 | Class / Attribute |
| SYSVNTWK.NSQLCOMMENT | 2 | Comment |
| SYSVNTWK.NSQLSCHEMA | 1 | Non SQL Schema |
| SYSVNTWK.NSQLSET | 2 | Set |
| SYSVNTWK.PRINTER | 3 | Printer in Destination |
| SYSVNTWK.PROGRAM | 2 | Program |
| SYSVNTWK.PTERM | 3 | Pterm |
| SYSVNTWK.PTERM_LTERM | 3 | Lterm in Line |
| SYSVNTWK.PTERM_LTERM | 4 | Lterm in Pterm |
| SYSVNTWK.QUEUE | 2 | Queue |
| SYSVNTWK.RCM | 1 | Relational Command Module |
| SYSVNTWK.RECORDELEM | 2 | IDD Record Element |
| SYSVNTWK.RECORDELEM | 3 | Record Element |
| SYSVNTWK.RECORDELEM | 4 | Record Element |
| SYSVNTWK.RECORDPROC | 3 | Data Compression Table |
| SYSVNTWK.RECORDPROC | 3 | Record Procedure |
| SYSVNTWK.RECORDPROC | 4 | Data Compression Table |
| SYSVNTWK.RECORDPROC | 4 | Record Procedure |
| SYSVNTWK.RECORDSHARE | 3 | Structure Shared By |
| SYSVNTWK.RECORDSYN | 3 | Record Synonym |
| SYSVNTWK.RECORDSYN | 4 | Record Synonym |
| SYSVNTWK.RECORDTYP | 2 | Record |
| SYSVNTWK.RECORDTYP | 3 | Network Record accessed |
| SYSVNTWK.RECORDTYP | 3 | Record in Area |
| SYSVNTWK.RECORDTYP_SET_ME | 3 | Set Record is member |
| SYSVNTWK.RECORDTYP_SET_OW | 3 | Set owned by Record |
| SYSVNTWK.RESTABLE | 2 | Destination Node in Res Table |
| SYSVNTWK.RESTABLE | 2 | Dictionary via Node in Res Table |
| SYSVNTWK.RESTABLE | 2 | Local Dictionary in Res Table |
| SYSVNTWK.RUNUNIT | 2 | Rununit |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|--------------------------------|
| SYSVNTWK. SET_RECORD_MEMBER | 3 | Member Record of Set |
| SYSVNTWK. SET_RECORD_OWNER | 3 | Owner Record of Set |
| SYSVNTWK.SET_REC_CONTROLKEY | 4 | Record Control Key |
| SYSVNTWK. SET_REC_FOREIGNKEY | 4 | Record Foreign Key |
| SYSVNTWK.STORPOOL | 2 | Storage Pool |
| SYSVNTWK.STORPOOL | 2 | XA Storage Pool |
| SYSVNTWK.SUBPROGAREA | 4 | Area used in Program |
| SYSVNTWK. SUBPROGAREA_NSQLAR | 5 | Area Statistics for Program |
| SYSVNTWK. SUBPROGAREA_NSQLAR | 5 | Schema definition of Area |
| SYSVNTWK. SUBPROGRECORD | 4 | Record used in Program |
| SYSVNTWK. SUBPROGRECORD_SUBR | 5 | Record Statistics for Program |
| SYSVNTWK. SUBPROGRECORD_SUBR | 5 | Subschema definition of Record |
| SYSVNTWK.SUBPROGSET | 4 | Set used in Program |
| SYSVNTWK. SUBPROGSET_NSQLSET | 5 | Schema definition of Set |
| SYSVNTWK. SUBPROGSET_NSQLSET | 5 | Set Statistics for Program |
| SYSVNTWK.SUBRECORD | 3 | Record in Subschema |
| SYSVNTWK.SUBRECORDELEM | 4 | Element of Subschema Record |
| SYSVNTWK.SUBSCHEMA | 2 | Subschema |
| SYSVNTWK.SUBSCHEMAATTRIB | 3 | Class / Attribute |
| SYSVNTWK. SUBSCHEMACOMMENT | 3 | Comment |
| SYSVNTWK.SUBSCHEMA_AREA | 3 | Area in Subschema |
| SYSVNTWK.SUBSCHEMA_PROG | 3 | Program using Subschema |
| SYSVNTWK.SUBSCHEMA_SET | 3 | Set in Subschema |
| SYSVNTWK.SYSTEM | 1 | System |
| SYSVNTWK.SYSTEM_LTERM | 2 | Lterm |

Table 4 — Resourcename in Alphabetic Order, Level and Objectname

| Resourcename (Schema.Table) | Level | Objectname |
|------------------------------------|--------------|-----------------------------------|
| SYSVNTWK.SYSTEM_PRINTER | 2 | Printer |
| SYSVNTWK.TASK | 2 | Task |
| SYSVNTWK.U_NSQLSHEMA | 2 | Non SQL Schema registered |
| SYSVNTWK.U_NSQLSHEMA | 2 | Non SQL Schema responsibility |
| SYSVNTWK.U_NSQLSHEMA | 2 | Registree for Schema |
| SYSVNTWK.U_NSQLSHEMA | 2 | Responsible for Schema |
| SYSVNTWK.U_SUBSCHEMA | 2 | Subschema registered |
| SYSVNTWK.U_SUBSCHEMA | 2 | Subschema responsibility |
| SYSVNTWK.U_SUBSCHEMA | 3 | Registree for Subschema |
| SYSVNTWK.U_SUBSCHEMA | 3 | Responsible for Subschema |
| SYSVSYST.AM_AMDEP | 3 | Access Module for Table |
| SYSVSYST.AM_AMDEP | 3 | Access Module for Table Procedure |
| SYSVSYST.AM_AMDEP | 3 | Access Module for View |
| SYSVSYST.AM_AMDEP | 3 | Table Procedure accessed |
| SYSVSYST.AM_AMDEP | 3 | Table accessed |
| SYSVSYST.AM_AMDEP | 3 | View accessed |
| SYSVSYST.CONSTRAINT | 2 | Constraint |
| SYSVSYST.CONSTRAINT | 3 | Constraint Table referenced |
| SYSVSYST.CONSTRAINT | 3 | Constraint Table referencing |
| SYSVSYST.CONSTRAINT | 3 | Referenced Table in Constraint |
| SYSVSYST.CONSTRAINT | 3 | Referencing Table in Constraint |
| SYSVSYST.DBNAME_SCHEMA | 3 | Schema referencing DBName |
| SYSVSYST.DMCL_AREA | 2 | Area in DMCL |
| SYSVSYST.DMCL_AREA | 3 | DMCL using Area |
| SYSVSYST.DMCL_FILE | 2 | File in DMCL |
| SYSVSYST.DMCL_FILE | 3 | DMCL in File |
| SYSVSYST.VDMCLAREA | 2 | Segment.Area Override |
| SYSVSYST.VDMCLFILE | 2 | Segment.File Override |

